ISSN 0253-3839

JOURNAL OF THE CHINESE INSTITUTE OF ENGINEERS

Vol. 21, No. 3
May 1998

Transactions of the Chinese Institute of Engineers, Series A

中國工程師學會學刊系列 A

中國工程學刊

Published by the Chinese Institute of Engineers,
Taipei, Taiwan, Republic of China.

中 國 工 程 學 刊

JOURNAL OF THE CHINESE INSTITUTE OF ENGINEERS

PUBLISHER: C.K. Shih（發行人：石中光）
PUBLISHED BY:
Chinese Institute of Engineers（發行所：中國工程師學會）
Address: #1, 4th Fl., Sec. 2, Jen-Ai Rd., Taipei, Taiwan 10019, R.O.C.（台北市仁愛路二段一號四樓）

Editor-in-Chief: C.T. Liou（總編輯：劉清田）
National Taiwan Univ. of Sci. and Tech.（國立台灣科技大學）
Address: 43, Sec. 4, Keelung Rd., Taipei, Taiwan 10672, R.O.C.
（台北市基隆路四段43號）

The Journal of the Chinese Institute of Engineers is published bimonthly. Institutional subscription rate: NT$2,400.00 annually; personal subscription rate: NT$1,800.00 for nonmembers, NT$1,000.00 for members. The annual rates for foreign subscriptions are US$120.00 for organizations and US$60.00 for individuals (including surface mail postage).（中國工程學刊每年出刊六次。國內機關學校訂閱每年新台幣二、四〇〇元；個人訂閱：非會員一、八〇〇元，會員一、〇〇〇元。國外機關學校訂閱每年美金一二〇元，個人六〇元（含陸海運郵資）。付款郵政劃撥帳號：〇〇〇五九八九～二號；戶名：中國工程師學會，電話：(02)2392-5128。）

*The publication of this Journal is partially subsidized by the National Science Council of the Republic of China.（本學刊之發行係由行政院國家科學委員會補助部份經費。）

Articles in the JOURNAL are indexed in the Engineering Index, SciSearch and Research Alert.

行政院新聞局出版事業登記證局版北市誌字第1068號

MULTIMEDIA SYNCHRONIZATION WITH USER INTERACTIONS USING INTERACTIVE EXTENDED FINITE STATE MACHINES (IEFSMS)

Chung-Ming Huang* and Chian Wang
Institute of Information Engineering
National Cheng Kung University
Tainan, Taiwan 701, R.O.C.

Key Words: Multimedia, Synchronization, Extended Finite State Machines (EFSMs), User interactions.

ABSTRACT

One of the main and required characteristics of multimedia systems is the user-interaction service. The user-interaction service is an essential requirement in some applications, e.g., Video-On-Demand (VOD) and News-On-Demand (NOD). The user-interaction service provides flexible multimedia presentations with user interactions. That is, users are allowed to have on-line adjustment of the presentation flow, e.g., skip some (boring) media units or reverse the presentation direction, to have some special features. In this paper, we propose an Interactive Extended Finite State Machine (IEFSM) model to specify synchronization issues in multimedia presentations with user interactions. By incorporating interrupt transitions and dynamic transitions in the IEFSM model, dynamic behaviors resulting from user interactions can be modeled using some IEFSMs. Using the IEFSM model, intra-medium synchronization is handled by an Actor, which is formally represented as an IEFSM; inter-media synchronization is handled by a Synchronizer, which is also formally represented as an IEFSM. The communication between IEFSMs is message-passing through some First-In-First-Out (FIFO) queues. In this way, the dynamic behaviors of user interactions, including reverse, skip, freeze-restart, and scale can be represented in IEFSM-based multimedia synchronization specifications.

I. INTRODUCTION

With the rapid progress and wide acceptance of computer-based applications, the need for multimedia systems is growing dramatically in a variety of fields, including business, manufacturing, education, Computer-Aided Design(CAD)/Computer-Aided Engineering(CAE), medicine, entertainment, etc. A multimedia system combines text, graphics, image, audio, video, and/or animation, to enhance information presentations [8, 9]. Because of the combination of several media streams, a multimedia presentation should deal with not only intra-media consistency issues but also inter-media consistency

*Correspondence addressee

issues. That is, a smooth multimedia presentation should maintain both intra-media synchronization and inter-media synchronization [3]. Essentially, there are two synchronization issues, i.e., temporal synchronization and spatial synchronization [15, 18]. Temporal synchronization maintains the temporal schedule of a multimedia presentation. Spatial synchronization maintains the spatial layout of displaying media units in a multimedia presentation. In this paper, we study the temporal synchronization issue for multimedia presentations with user interactions. These inte-ractions include reverse, skip, freeze-restart, and (re-)scale of the presentation speed.

Essentially, media can be classified into two types: static media and continuous media. Static media have no temporal properties within themselves. Static media include still images, text, and graphics. The presentation durations of static media are synthetically determined by applications. Continuous media contain sequences of media units. Audio, video, and animation belong to continuous media. Presentation durations of continuous media are embedded.

There are two types of multimedia synchronization, i.e., intra-medium synchronization and inter-media synchronization. Intra-medium synchronization affects the rates of the presentations. Inter-media synchronization deals with maintaining the requirements of temporal relationships among media streams, such as lip synchronization between video and audio. A lot of control schemes and formal models have been proposed to achieve intra-medium and inter-media synchronization controls [2, 5, 6, 10, 12, 14, 15, 20, 21]. The modeling of user interactions becomes complicated in multimedia presentations because it should deal with issues in (1) both intra-medium and inter-media synchronization, and (2) the processing of dynamic user-interactions, which are issued unpredictably. Several schemes have been proposed to control multimedia synchronization with user interactions [11, 13, 17], and several models have been used to specify multimedia synchronization with user interactions [7, 16, 19]. One type of approach is to represent synchronization relationships using programming-language-like paradigms [7, 19]. Using the programming-language-like approach, all temporal events are described as a set of procedures. Temporal events in these procedures are executed either sequentially or in parallel. Another type of approach is to use state-transition models to describe the control flow of composed components that are involved in the presentations [16]. In [16], the Augmented Object Composition Petri Nets (AOCPN) model is proposed. This model has formal specification and modeling of multimedia synchronization with user interaction. Programming-language-like approaches can specify dynamic behaviors of multimedia synchronization, but the representation of control flow is not explicit. On the contrary, the AOCPN model can specify the control flow very clearly. But the AOCPN model is not capable enough to specify (1) the dynamic behaviors of multimedia synchronization, and (2) the re-synchronization actions that are adopted to eliminate asynchronous anomalies. Thus, the AOCPN model can specify the static configuration of multimedia synchronization with user interactions [16].

The Finite State Machine (FSM) model has been widely used in formal modeling. For example, the Communicating Finite State Machine (CFSM) [4] model has been used in the formal modeling of communication protocols. In [12], our research colleagues have used the Extended Finite State Machines (EFSMs) to have formal specifications of multimedia synchronization control. By extending the EFSM, we propose a hybrid model in this paper. This model is called the Interactive Extended Finite State Machine (IEFSM) model, to specify multimedia synchronization with user interactions.

The IEFSM model contains two parts: the state-transition part and the programming language part. The state-transition part provides the capability of explicitly describing synchronization control flow; while the programming language part, which contains variables and operations on the variables, provides the capability of describing dynamic behaviors in multimedia synchronization. That is, the programming language part can assist the representation of the state-transition part. As a result, the dynamic configuration contained in the multimedia synchronization with user interactions can be modeled using IEFSMs. Additionally, some asynchrony anomalies may exist at any time, e.g., when users interrupt the presentation stream. Thus, multimedia presentations need to be re-synchronized after the processing of user interactions is finished. That is, some re-synchronization actions should also be adopted in the processing of user interactions. Using the proposed IEFSM model, the re-synchronization actions can also be modeled.

There are two types of IEFSMs in IEFSM-based specifications of multimedia synchronization with user interactions: *Synchronizer* IEFSMs and *Actor* IEFSMs. A Synchronizer IEFSM controls inter-media synchronization. An Actor IEFSM controls intra-medium synchronization. With the cooperation and the coordination of Synchronizer and Actor IEFSMs, synchronous multimedia presentations with user interactions, e.g., reverse, skip, freeze-restart, and scale, can be formally modeled.

The rest of this paper is organized as follow.

ISSN:0253-3839

Journal of the Chinese Institute of Engineers

May 1998 Volume 21, No. 3

CONTENTS

Papers

Short Papers

台灣郵政管理局雜誌交寄
執照台字第4294號

行政院新聞局出版事業登記證
局版北市誌字第1068號

中　國　工　程　學　刊

民國八十七年五月　　第二十一卷第三期

目　　錄

論文

短篇論文

Section 2 introduces the formal definition of the IEFSM model. Section 3 presents formal specifications of temporal relationships between two media objects using the IEFSM model. Section 4 presents an example of an IEFSM-based multimedia synchronization specification that has no user interaction. Section 5 points out the main synchronization issue for processing user interactions. Section 6 presents multimedia synchronization with user interactions using IEFSMs. Section 7 has concluding remarks and directions for future work.

II. THE IEFSM MODEL

In the Interactive Extended Finite State Machine (IEFSM) model, an IEFSM is defined as a nine-tuple $(\Sigma, S, s_0, V, E, D, P, A, \delta)$, where

- Σ is the set of messages that can be sent or received,
- S is the set of states,
- s_0 is the initial state,
- V is the set of variables,
- E is the set of predicates that operate on variables,
- D is the set of delay classes, which specify some time constraints,
- P is the set of priority clauses,
- A is the set of actions that operate on variables,
- δ is the set of state transition functions, where each state transition function is formally represented as follows: $S\times\Sigma\times E(V)\times P\times D\rightarrow \Sigma\times A(V)\times S$.

In the IEFSM model, a state transition is denoted as "$S_1 \xrightarrow{T} S_2$",which means that an IEFSM executes transition T at state S_1 and then enters into state S_2; T is called an outgoing transition of S_1 and an incoming transition of S_2; and S_1 is called the head state of T, and S_2 is called the tail state of T. Given a "$S_1 \xrightarrow{T} S_2$", the IEFSM remains in its head state S_1 during the execution of T; when the execution of T is finished, the IEFSM's state is then changed to the tail state S_2. There are two parts in a transition: the condition part and the action part. The condition part can contain (1) an input event, (2) a predicate, which is a boolean expression, (3) a time clause, which is represented as "delay[t_{min}, t_{max}]", and (4) a priority clause, which is represented as "priority *scale*". The action part can contain output events and a number of statements that operate on variables. If the execution of a transition T has some time constraints, a time clause is associated with T. The time clause of a transition identifies when the transition should be executed if the transition is executable. The priority clause of a transition specifies the execution priority of the transition. If there is no priority clause, the priority is middle. A transition T can be executed when (1) the input event is available, (2) the predicate is true, (3) its priority is the highest among executable transitions at the IEFSM's current state. If T is associated with a time clause, let the time clause be $[t_{min}, t_{max}]$. When transition T is executable and T's priority is the highest among the executable transitions, T should be executed between time $t+t_{min}$ and time $t+t_{max}$, where t is the time the associated IEFSM enters into T's head state. If t_{min} is equal to t_{max}, it means that T should be executed at time $t+t_{min}$. Transitions are classified into two types: (1) spontaneous transitions: the transitions whose condition parts have no input events, and (2) when transitions: the transitions whose condition parts have input events. The execution of a transition can be atomic or un-atomic. An atomic transition, which is denoted as AT, cannot be interrupted; an un-atomic transition, which is denoted as T, can be interrupted and terminated.

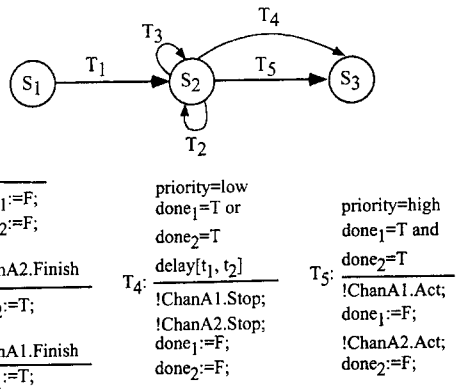


Fig. 1. An example of of an IEFSM.

IEFSMs communicate with each other via message passing through a number of First-In-First-Out (FIFO) bidirectional communication queues. Fig. 1 illustrates an example of an IEFSM[1], where a circle represents a state, an arc represents a transition,"? *chan.m*" represents that message m is input from channel chan, "! *chan.m*" represents that message m is output to channel *chan*, "delay[t_1, t_2]" represents the time clause, and "priority n" represents the priority clause. There are three states and five transitions in Fig. 1. Transitions T_1, T_4, and T_5 are spontaneous transitions, and transition T_2 and T_3 are when transitions. Whenever the IEFSM enters into state S_1, transition T_1 is executed. When the execution of T_1 is finished, the IEFSM's current state is changed to state S_2. The priority clauses in transitions T_4 and T_5 are used to decide which transition can be executed when more

[1]For simplicity, T represents TRUE and F represents FALSE in the figures of this paper.

than one transition are executable. Transition T_4 has a delay clause "delay[t_1, t_2]", so T_4 can be executed between time [$t+t_1$, $t+t_1$] when its predicate "*done1= TRUE* or *done2=TRUE*" is true, where t is the time the IEFSM enters into state S_2. But if both transitions T_4 and T_5 are executable, transition T_5 is selected for execution because it has a higher priority than that of T_4.

During a multimedia presentation, the user may want to control the sequence of the presentation flow. Typical user interactions include *reverse*, *skip*, *freeze-restart*, and scale. That is, users may (1) *reverse* the direction of the presentation flow, (2) *skip*, either forwardly or backwardly, several media units or even some media stages/objects, (3) *freeze* the presentation flow, and then *restart* the presentation flow after some time units, and (4) (re-) *scale* the presentation speed, either speed-up or slow-down. In the proposed IEFSM model, synchronization mechanisms are provided for describing user interactions to modify the presentation flow. Since user interactions are applied to on-execution presentations, the corresponding user inputs are treated as interrupts to the associated IEFSMs. That is, currently executing transitions must be pre-emptied. Additionally, users may skip to any one of the presentation points, the tail states of the transitions dealing with the corresponding user inputs are thus dynamic. Therefore, the IEFSM model provides two kinds of special transitions, which are called interrupt transitions (*ITs*) and dynamic transitions (*DTs*).

In the IEFSM model, the pre-emption is handled by *interrupt transitions* (*ITs*). An input message which resulted from a user interaction, i.e., a user input, invokes the execution of an interrupt transition. That is, interrupt transitions will be executed when the associated IEFSMs receive some user interactions from user input channels. *ITs* can interrupt the execution of un-atomic transitions[2]. *ITs* themselves are atomic transitions.

Some user interactions, e.g., the skip operation, which allows users to specify a new starting point of a presentation, cause dynamic state changes in the associated IEFSMs. *Dynamic transitions* (*DTs*) are used to handle this requirement. The tail state of a *DT* is decided at the run time. In other words, at the specification stage, the tail state of a dynamic transition DT_d becomes a variable, and the action part of DT_d contains some statements that are used to derive the tail state after the execution of DT_d. An example is as follows: Let DT_d be a dynamic transition and $S(i) \xrightarrow{DT_d} S(j)$. The relationship of S_i and S_j can be

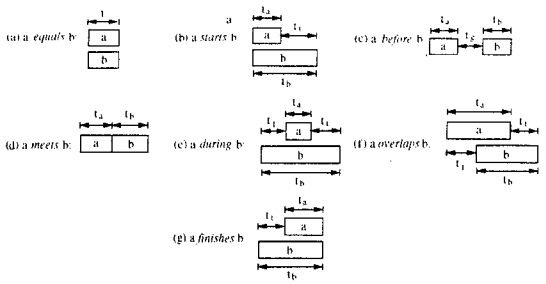


Fig. 2. Possible temporal relationships between two objects.

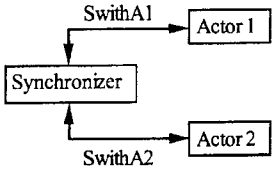
$j=i+k*x$, where x is a parameter contained in the user input message and k is a coefficient. *DTs* are interrupt transitions, and are thus atomic transitions.

III. TEMPORAL RELATIONSHIPS IN THE IEFSM MODEL

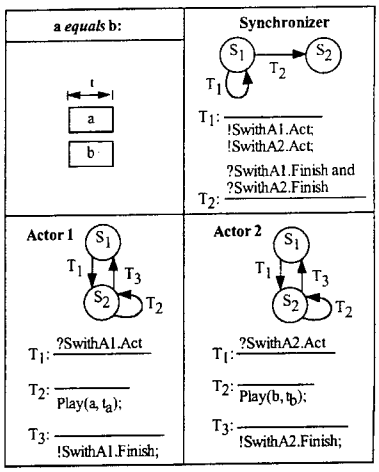
A multimedia presentation integrates several media streams, which have different temporal characteristics, to present information to users. Temporal relationships among media streams can be represented by *temporal intervals* [15]. Given any two intervals, there are thirteen different ways in which the two intervals may be related [1]. Fig. 2 depicts seven of the thirteen relationships; the remaining ones are the inverse of relations (b)...(g) that are depicted in Fig. 2.

Figure 3 depicts the formal specifications of the seven temporal relationships using the IEFSM model, in which Fig. 3-(a) depicts the configuration of channel connections. The seven temporal relationships depicted in Figs. 3-(b)...3-(h) can be grouped into four groups. (i) Equal and start: With these two relationships, the presentation of objects *a* and *b* starts at the same time. Thus, the Synchronizer sends two *Act* messages to Actors 1 and 2 to commence the presentation. (ii) Before: Object *b* has to wait for t_g time units after object *a* finishes its presentation. The delay clause *delay*(t_g, t_g) in transition T_2 of Actor 2 handles the temporal gap. (iii) Meet: Object *b* starts its presentation immediately after the presentation of object *a* is finished. Thus, when the Synchronizer receives the *Finish* message sent from Actor 1, the Synchronizer sends an *Act* message to Actor 2 to commence object *b's* presentation. (iv) During and finish (overlap): The presentation of object *a* (*b*) has to wait for t_f time units after the presentation of object *b* (*a*) is commenced. The delay clause *delay*(t_f, t_f) in transition T_2 of Actor 1 (2) handles the temporal gap.

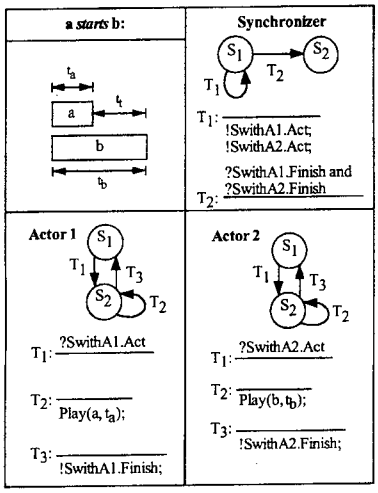
[2] A state S of an IEFSM I can accept user interactions when (1) I is at state S, and I is either (i) without executing any transition, or (ii) executing an un-atomic transition, and (2) S's outoging transitions have the corresponding interrupt transitions; otherwise, the user interactions will not be accepted.


SwithA1
Actor 1
Synchronizer
Actor 2
SwithA2

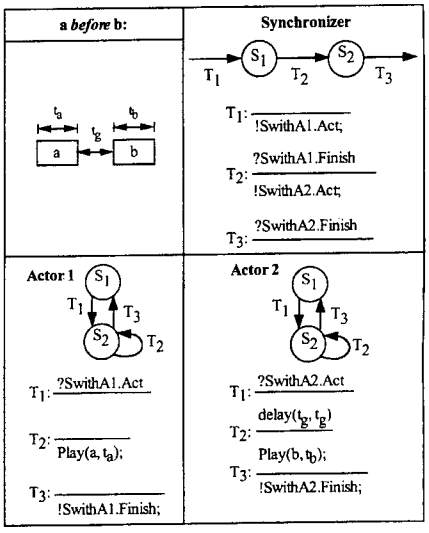
(a)


a equals b:
Synchronizer
T1: !SwithA1.Act; !SwithA2.Act;
T2: ?SwithA1.Finish and ?SwithA2.Finish
Actor 1
T1: ?SwithA1.Act
T2: Play(a, ta);
T3: !SwithA1.Finish;
Actor 2
T1: ?SwithA2.Act
T2: Play(b, tb);
T3: !SwithA2.Finish;

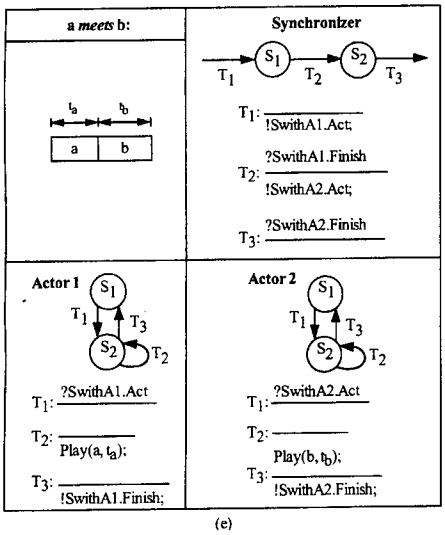
(b)


a starts b:
Synchronizer
T1: !SwithA1.Act; !SwithA2.Act;
T2: ?SwithA1.Finish and ?SwithA2.Finish
Actor 1
T1: ?SwithA1.Act
T2: Play(a, ta);
T3: !SwithA1.Finish;
Actor 2
T1: ?SwithA2.Act
T2: Play(b, tb);
T3: !SwithA2.Finish;

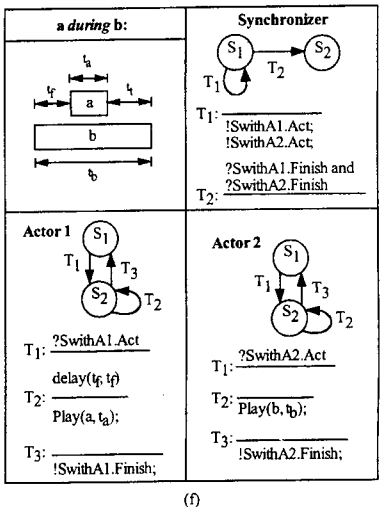
(c)


a before b:
Synchronizer
T1: !SwithA1.Act;
T2: ?SwithA1.Finish !SwithA2.Act;
T3: ?SwithA2.Finish
Actor 1
T1: ?SwithA1.Act
T2: Play(a, ta);
T3: !SwithA1.Finish;
Actor 2
T1: ?SwithA2.Act
T2: delay(tg, tg) Play(b, tb);
T3: !SwithA2.Finish;

(d)


a meets b:
Synchronizer
T1: !SwithA1.Act;
T2: ?SwithA1.Finish !SwithA2.Act;
T3: ?SwithA2.Finish
Actor 1
T1: ?SwithA1.Act
T2: Play(a, ta);
T3: !SwithA1.Finish;
Actor 2
T1: ?SwithA2.Act
T2:
T3: Play(b, tb); !SwithA2.Finish;

(e)


a during b:
Synchronizer
T1: !SwithA1.Act; !SwithA2.Act;
T2: ?SwithA1.Finish and ?SwithA2.Finish
Actor 1
T1: ?SwithA1.Act
T2: delay(tf, tf) Play(a, ta);
T3: !SwithA1.Finish;
Actor 2
T1: ?SwithA2.Act
T2: Play(b, tb);
T3: !SwithA2.Finish;

(f)

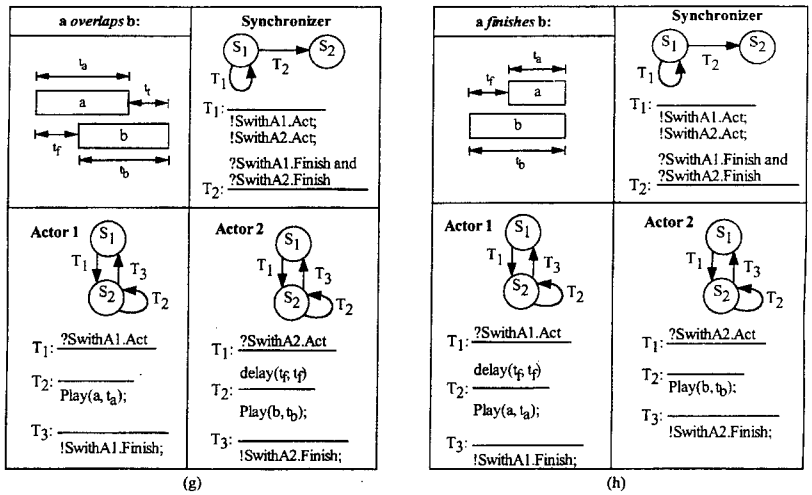


Fig. 3. Formal specifications of the seven temporal relationsips using the IEFSM model: (a) channel configuration, (b) equal relationship, (c) start relationship, (d) before relationship, (e) meet relationship, (f) during relationship, (g) overlap relationship, and (h) finish relationship.

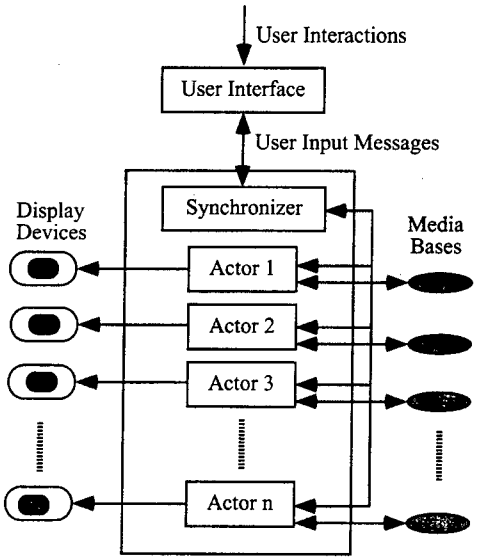


Fig. 4. The abstract architecture of IEFSM-based multimedia synchronization with user-interactions.

IV. MULTIMEDIA SYNCHRONIZATION USING IEFSMS

In this Section, we use the IEFSM model to formally specify multimedia synchronization without considering user interactions. Based on the hybrid characteristic of the IEFSM model, the synchronization control part and the dynamic data variables part can be specified. In a formal synchronization specification, the behavior of a data stream is represented as an IEFSM, which is called Actor IEFSM.

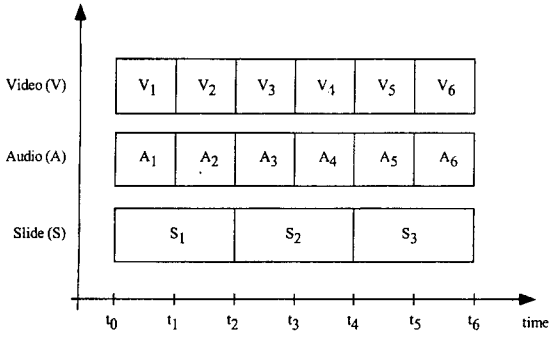


Fig. 5. An example of a multimedia presentation.

Inter-stream synchronization among media streams is also represented as an IEFSM, which is called the Synchronizer IEFSM. The Synchronizer holds information about temporal relationships among all streams. Each Actor denotes one stream and controls the data flow of the associated stream. That is, intrastream synchronization is achieved in an Actor, and in-terstream synchronization is achieved in the Synchronizer. The communication between the Synchronizer IEFSM and Actor IEFSMs are message-passing through some FIFO queues. Fig. 4 shows the abstract architecture of IEFSM-based multimedia synchronization with user interactions.

For convenience, the multimedia presentation depicted in Fig. 5 is used as an example for explanation. In Fig. 5, there are three media streams: a video stream (V), an audio stream (A), and a slide (image) stream (S). The example contains six presentation stages in video and audio streams, and three stages in the slide stream. Thus, seven time stamps, t_0, t_1, t_2,

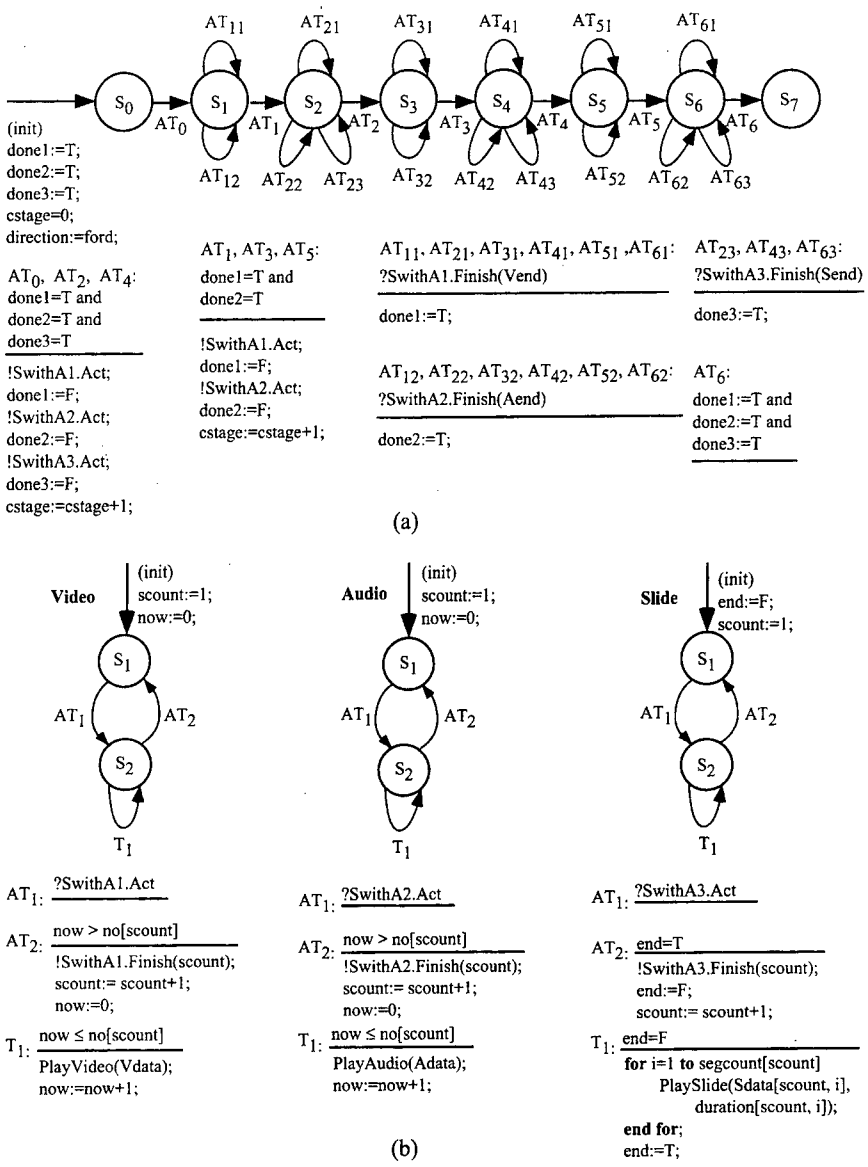


Fig. 6. (a) Synchronizer IEFSM, and (b) Actor IEFSMs, for the presentation that is depicted in Figure 5.

t_3, t_4, t_5, and t_6, which are called synchronization points, are required. Synchronization points represent the time points where two or more media have to start and/or end isochronously in the time axis. Accordingly, there are three Actor IEFSMs and a Synchronizer IEFSM. Fig. 6 depicts the corresponding Synchronizer and Actor IEFSMs. Each synchronization point is denoted by a state in the Synchronizer IEFSM. In the Synchronizer IEFSM, which is depicted in Fig. 6-(a), variables $done_1$, $done_2$, and $done_3$ denote the presentation status of the *V*, *A*, and *S* streams, respectively. Variable *cstage* denotes the current presentation stage. Variable *direction* denotes the flow direction, which is represented as *ford* in default. The value of *direction* becomes *back* if the flow direction is backward. The Synchronizer IEFSM communicates with video, audio, and slide Actor IEFSMs using channels *SwithA1*, *SwithA2*, and *SwithA3* respectively. The Synchronizer IEFSM invokes a synchronous commencement of a presentation stage by sending messages "Act" to Actor IEFSMs in transitions AT_0 to AT_5. The end of a presentation stage is denoted by receiving messages "Finish", which are sent from Actor IEFSMs, e.g., in

transitions AT_{11} and AT_{12} at state S_1, transitions AT_{21}, AT_{22}, and AT_{23} at state S_2, etc. Variables *Vend*, *Aend*, and *Send* contain the currently finished presentation stages of the video, audio, and slide Actor IEFSMs respectively.

Figure 6-(b) shows the corresponding Actor IEFSMs. When an "Act" message is received, Actor IEFSM starts to present media objects. In the action parts of Actor IEFSMs, procedures *PlayVideo*, *PlayAudio*, and *PlaySlide* contain the corresponding system routines to display the associated media objects. Actors for different media types have different duration management schemes, depending on the nature of the media. Actors of continuous media, such as video and audio, need not to specify the durations because continuous media contain their own temporal constraints; while Actors of static media, such as the slide stream, must specify the presentation durations. The specification of presentation durations is achieved by using an additional parameter, i.e., *duration*[*scount*], in the PlaySlide procedure. The variable array *segcount*[*scount*] keeps the number of segments of each stage for the slide stream.

Figure 7 depicts the corresponding interaction sequence for the IEFSMs depicted in Fig. 6. When a multimedia presentation is invoked, Actors wait for the "Act" messages sent from the Synchronizer to have the commencement of displaying media objects. In the video (audio) Actor IEFSM, which is depicted in Fig. 6-(b), *no*[*scount*] contains the number of frames (segments) of the *scount*th presentation stage, and variable *now* denotes the number of displayed frames (segments). In slide Actor IEFSM, the duration of the current presentation stage is denoted by a parameter, i.e., *duration*[*scount*], in the *PlaySlide* procedure. At the end of each presentation stage, each Actor IEFSM sends a "Finish" message to the Synchronizer IEFSM to notify the end of the current stage. When the Synchronizer IEFSM receives all "Finish" messages sent from Actor IEFSMs, the Synchronizer IEFSM advances to the next state and sends "Act" messages to Actor IEFSMs to start the next presentation stage.

The predicate mechanism is an important feature in the IEFSM model. Without the predicate mechanism, which is the situation in the pure Finite State Machine (FSM) model, specifications of Actors and the Synchronizer become very complicated. A corresponding FSM-based specification for the multimedia presentation depicted in Fig. 5 is given in the Appendix.

V. THE MAIN SYNCHRONIZATION ISSUE FOR PROCESSING USER INTERACTIONS

According to different media properties, each medium can adopt different intra-medium synchronization policies when random processing delays occur. For example, to keep visual continuity, video streams can adopt the non-blocking synchronization policy, in which the most recently displayed medium unit is repeatedly displayed until the expected medium unit arrives [12]. However, because it is nonsense to re-display a segment of audio repeatedly, audio streams can adopt the blocking synchronization policy, in which the display is blocked until the expected medium unit arrives [12]. Because of the adopted synchronization policies, the displaying media units of different media streams may not be the originally coupled ones when user interactions are issued and the presentation is interrupted. That is, the display of media streams may not be synchronous when a user's interaction input is received and each medium stream's presentation is interrupted and paused temporarily.

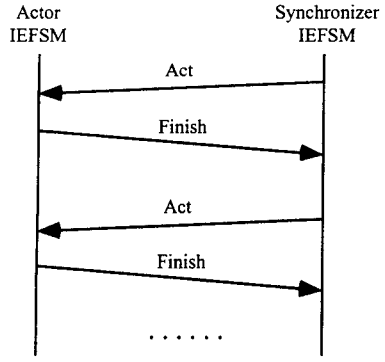


Fig. 7. The interaction sequence for Figure 6.

In order to process user interactions conveniently, the concept of "flow index" can be adopted. The flow index is a reference index from a given presentation point to the initial presentation point. Each medium stream is associated with a flow index counter. The flow index counter is similar to the tape index counters used in our home VCRs. A unit of a flow index can be one or multiple atomic media units, in which an atomic medium unit is an inseparable unit of the video medium. For example, an atomic medium unit for video streams is a frame. In this paper, the x^{th} flow index of medium stream S represents the x^{th} medium unit of S. Thus, when a medium unit of S is repeatedly displayed, S's flow index remains unchanged. The display length of a flow index unit is the time duration of a video frame. The time length of a medium's flow index unit is equal to the others', but the data size may be different from others'. Fig. 8 depicts an illustrative example when the presentation is interrupted and paused temporarily. Due to random processing

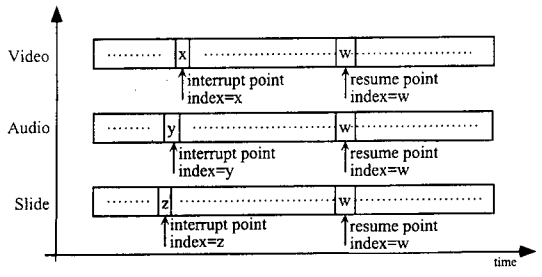


Fig. 8. The asynchronous phenomenon when a user interaction issued.

delays, flow indices of video, audio, and slide streams are x, y, and z respectively when the user issues an interactive function. After processing the interactive function, resume points of all media streams should be the same, e.g., w, which is derived by using a re-synchronization function $F_I(x, y, z)$, where I is the interaction type. Thus, the presentation will be resumed synchronously.

The re-synchronization issue is resolved by adopting different re-synchronization functions F_I, where I is skip, reverse, scale, and freeze-restart. Let medium stream i be in flow index x_i, i=1..n, when the presentation is temporarily paused. For the skip interactive function, since the destination points are specified by users, e.g., each medium stream skips to the x^{th} medium unit, the presentation is resumed synchronously from medium unit x, no matter what the values of $x_1, x_2, ..., x_n$ are. That is, $F_{skip}(x_1, x_2, ..., x_n)=x$ for the skip interactive function.

For reverse, scale, and freeze-restart interactive functions, presentations should also be resumed synchronously. In order to have a synchronous resumed presentation in the IEFSM model, we have slower streams keep pace with the fastest stream. Thus, in the forward presentation case, the new flow index is set to the maximum value of media streams' interrupted flow indices, i.e., $F_I(x_1, x_2, ..., x_n)=$ $Maximum(x_1, x_2, ..., x_n)$, I is scale or freeze-restart. In the backward presentation case, the new flow index is set to the minimum/maximum value of media streams' interrupted flow indices, i.e., $F_I(x_1, x_2, ..., x_n)=Minimum(x_1, x_2, ..., x_n)$, I is scale or freeze-restart. That is, the re-synchronization function F_I is to have slower streams skip some media units to keep pace with the fastest stream.

In the forward to backward presentation case, the new flow index is set to the maximum value of media streams' interrupted flow indices, for the reverse interactive function, i.e., $F_{reverse}(x_1, x_2, ..., x_n)=Minimum(x_1, x_2, ..., x_n)$. In the backward to forward presentation case, the new flow index is set to the minimum value of media streams' interrupted flow indices for the reverse interactive function, i.e., $F_{reverse}(x_1, x_2, ..., x_n)=Minimum(x_1, x_2, ..., x_n)$. That is, the re-synchronization function $F_{reverse}$ is to have slower streams skip some media units to keep pace with the fastest stream.

VI. PRESENTATIONS WITH USER INTERACTIONS

In this Section, we present multimedia synchronization with user interactions using the IEFSM model. Four user interactions that are adopted as the illustration are reverse, skip, freeze-restart, and scale.

1. The reverse operation

The purpose of a reverse operation is to reverse the presentation flow direction, i.e., from the forward (backward) direction to the backward (forward) direction. When a user wants to change the direction of a multimedia presentation flow, the reverse operation can be invoked. Some states and transitions should be modified/added in the original synchronization specifications without user interactions, e.g., the specification depicted in Fig. 6, to control the reverse operation. Based on the presentation flow depicted in Fig. 5, some states and transitions are added/ modified in the IEFSMs depicted in Fig. 6 to deal with the reverse operation. Fig. 9 depicts the modified Synchronizer IEFSM, Fig. 10-(a) depicts the modified Actor IEFSM for the video stream, and Fig. 10-(b) depicts the modified Actor IEFSM for the slide stream. The modified Actor IEFSM for the audio stream is similar to the modified Actor IEFSM for the video stream. Thus, for simplicity, the modified Actor IEFSM for the audio stream is not depicted. Additionally, Figs. 9 and 10 don't depict un-modified transitions because they can be referred in Fig. 6, i.e., only the modified/added transitions and states are depicted.

In Fig. 9, (1) a complement set of states, i.e., $\overline{S_i}$, and two complement sets of transitions, i.e., $\overline{AT_i}$ and $\overline{AT_{ij}}$, which control the backward presentation flow synchronization, and (2) a supplement set of states, i.e., S'_i, and some sets of transitions ${}_rIT_i$ and ${}_r\overline{IT_i}$, ${}_{rq}AT_i$ and ${}_{rq}\overline{AT_i}$, AT'_{ij} and $\overline{AT'_{ij}}$, and ${}_qAT_{ij}$, which are used for processing user inputs and re-synchronization when presentations are interrupted by users, are added. In the backward presentation case, (1) the Synchronizer sends messages "RAct" to Actors to commence the start of a backward presentation stage, (2) variable *now* is equal to *no*[*scount*] at the commencement, (3) variable *now* becomes 0 when a presentation stage is over, and (4) an Actor sends message "RFinish" to the Synchronizer to indicate the end of a backward presentation stage.

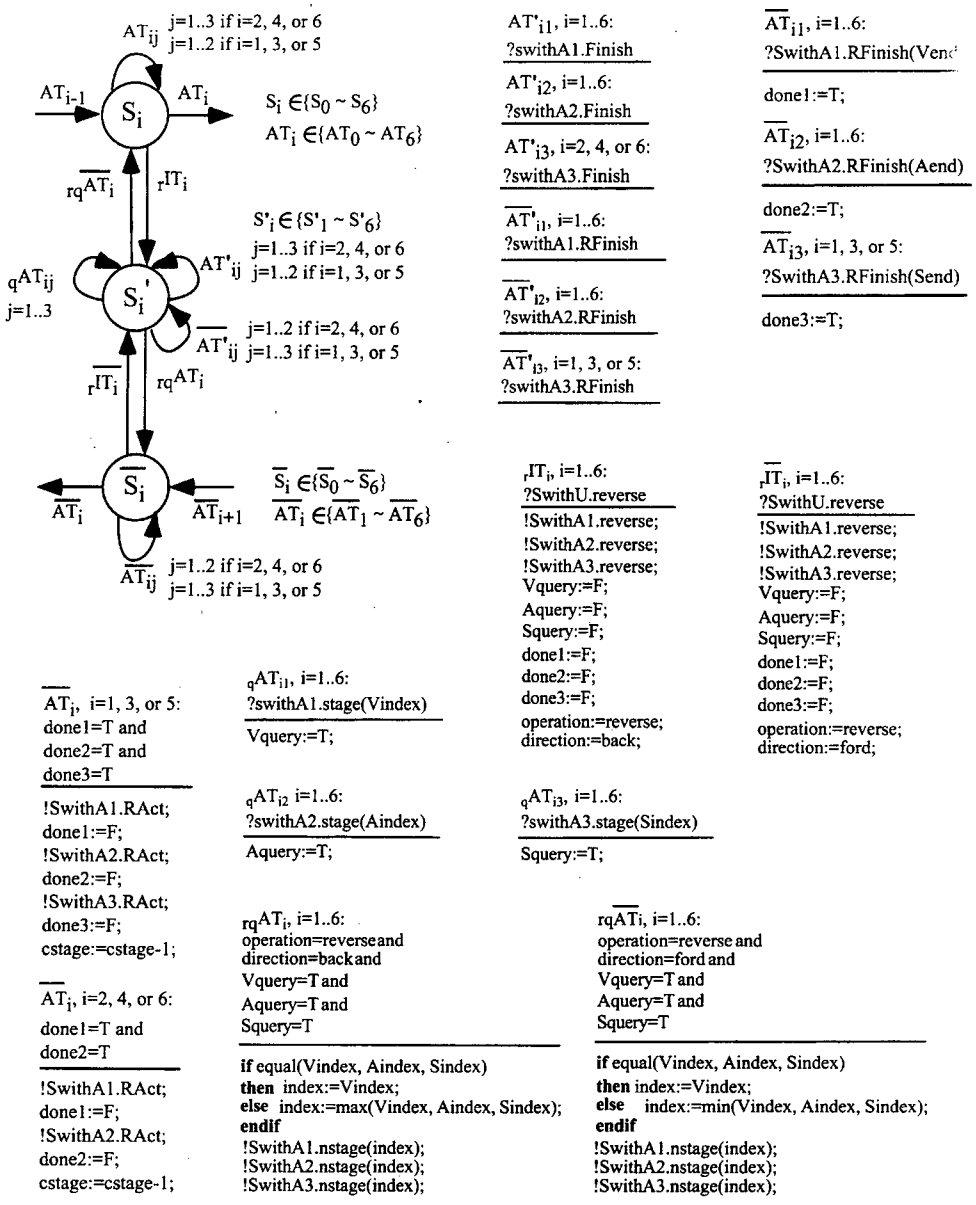


Fig. 9. Reverse state transitions of the Synchronizer.

Figure 11 depicts the interaction sequence for the reverse operation. When the Synchronizer receives a reverse operation from users in the forward (backward) flow direction, an interrupt transition $_rIT_i$ ($_r\overline{IT}_i$), i=1..6, is executed, depending on the current state of Synchronizer. The *SwithU* channel that is in transitions $_rIT_i$ and $_r\overline{IT}_i$ can accept user input messages. That is, the user can execute user interactions using some system-provided user interface, and then the system transforms each user interactions to the corresponding user input message. Interrupt transitions $_rIT_i$ ($_r\overline{IT}_i$), i=1..6, can receive *reverse* messages and output *reverse* messages to Actor IEFSMs in the forward (backward) presentation situation. Since some asynchronous anomalies may exist at any instant of a presentation, a query processing is invoked to decide whether these streams are currently synchronous or not. The query of synchronous situation can be achieved by adopting flow indices. When an Actor IEFSM X receives the *reverse*

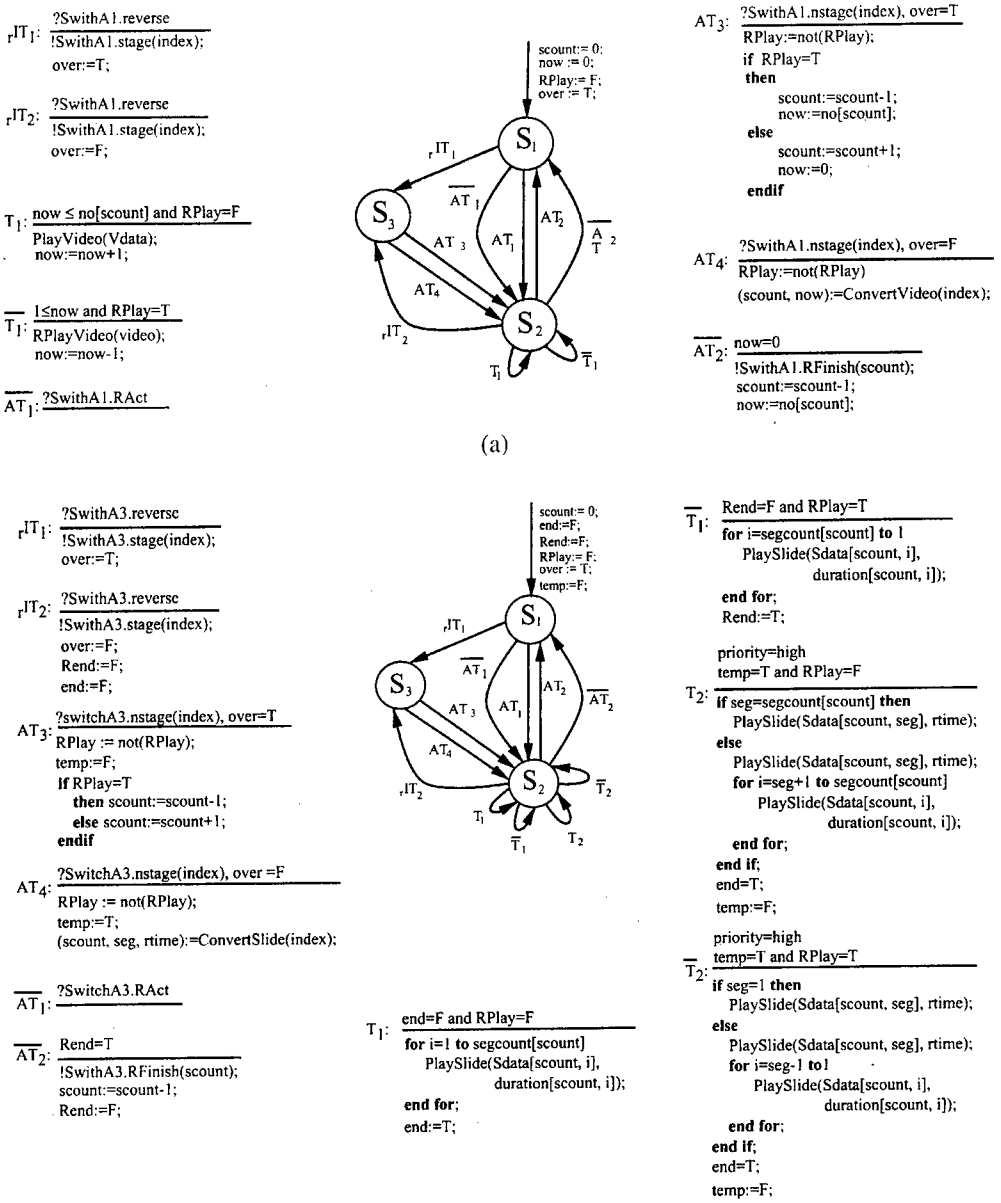


Fig. 10. Modified (a) video and (b) slide Actor IEFSMs for reverse operations.

message, X should send a message *stage()*, which contains X's current presentation flow index, to the Synchronizer IEFSM. At state S_i', i=1..6, three atomic transitions ${}_qAT_{ij}$, j=1..3, are used to receive *stage* messages that are sent from Actor IEFSMs. A message *stage* contains the current presentation flow indices of a medium stream. When the query processing of the current presentation flow indices is achieved, i.e., after all media's current presentation flow indices having been received, atomic transitions ${}_{rq}AT_i$ (${}_{rq}\overline{AT}_i$), i=1..6, can be executed. In transition ${}_{rq}AT_i$ (${}_{rq}\overline{AT}_i$), i=1..6 (*rq* represents reverse query), function *equal* decides whether these three streams' presentations are fully synchronous or not. Depending on whether the answer is (1) positive or (2) negative, which may result from different bus delays or different disk I/O delays for processing different media, the value of variable *index* is set to (1) the value of *Vindex*, because *Vindex=Aindex=Sindex*, or (2) the maximum (minimum) value of

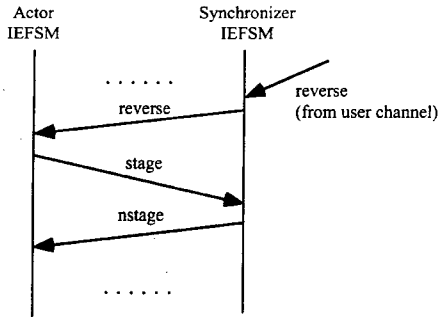


Fig. 11. The interaction sequence for the reverse operation.

variables Vindex, *Aindex*, and *Sindex* in the forward to backward (backward to forward) presentation situation. That is, the slower streams skip some media units or some display time to keep pace with the fastest stream. Then, message *nstage*, which contains the value of *index*, is sent to Actor IEFSMs.

Figure 10-(a) depicts the modified Actor IEFSMs for the video stream, and Fig. 10-(b) depicts the modified Actor IEFSM for the slide stream. A new state, i.e., S_3 is added. In Fig. 10, variable "RPlay" denotes the current presentation flow direction, RPlay is TRUE (FALSE) when it is in the backward (forward) presentation situation. Transition T_1 is modified and transition $\overline{T}_1$ (and T_2 and $\overline{T}_2$) is added, in order to handle the forward and backward presentation respectively in the video and audio streams (slide stream). Atomic transitions $\overline{AT}_1$ and $\overline{AT}_2$ are added to handle the commencement and end of a backward presentation stage respectively.

Whenever the Synchronizer receives reverse operations from users, it sends *reverse* messages to Actors, which triggers interrupt transition ${}_rIT_1$ or ${}_rIT_2$ in the Actor IEFSMs. Depending on the status of Actors, Actors may be at S_1 or S_2 when the *reverse* message is received. When an Actor X is at S_1, it means that X has finished its current presentation stage and is waiting for the commencement of the next presentation stage. When an Actor X is at S_2, it means that X is displaying media units that belong to the current presentation stage. When an Actor receives the "reverse" message, interrupt transition ${}_rIT_1$ or ${}_rIT_2$ is executed: the current presentation flow index is sent to the Synchronizer, and variable *over* is set to be TRUE or FALSE. That is, variable *over* denotes the status of an Actor IEFSM X: if X is at state S_1, *over* is TRUE; if X is at state S_2, *over* is FALSE.

As mentioned previously, the Synchronizer decides the synchronization situation of current presentation using the *equal* function in transitions ${}_{rq}AT_i$ (${}_{rq}\overline{AT}_i$), which are depicted in Fig. 9. Actors receive the new flow index using atomic transitions AT_3 (AT_4) when interrupt transition ${}_rIT_1$ (${}_rIT_2$) is executed at state S_1 (S_2). Each time the user issues a reverse operation, the current presentation flow direction will be reversed. That is, a backward presentation becomes a forward one, and vice versa. Whenever an Actor receives the "nstage" message from the Synchronizer, the "RPlay:=not(RPlay)" statement in transitions AT_3 and AT_4 of the Actor IEFSMs is executed to denote the following presentation flow direction. The following presentation after receiving a reverse operation is analyzed as follows:

In video and audio Actor IEFSMs, depending on the value of variable *over*, i.e., the interrupt transition is executed at state S_1 or S_2, and the possible situations are as follows:

- If *over* is TRUE, it means that (1) IEFSM is at state S_1 when the *reverse* message is received, and (2) the executed interrupt transition is ${}_rIT_1$. If the reverse situation is from forward to backward (backward to forward), i.e., variable *RPlay* becomes TRUE (FALSE), the value of variable *scount*, which records the following presentation stage, should be decreased (increased) by 1. The reason is that the value of *scount* has already increased (decreased) by 1 in transition AT_2 ($\overline{AT}_2$), which indicates the next presentation stage to be presented in the forward (backward) presentation case. Additionally, variable *now* is set to be *no*[*scount*] (0) in the forward to backward (backward to forward) case.
- If *over* is FALSE, it means that (1) the IEFSM is at state S_2 when the *reverse* message is received, and (2) the executed interrupt transition is ${}_rIT_2$. Since some asynchrony anomalies may exist, the flow index should be adjusted. Based on the (new) flow index, which is recorded in variable *index*, function *ConvertVideo*(*index*), which is *ConvertAudio*(*index*) in the audio Actor IEFSM, calculates the corresponding values of variables *scount* and *now* for the following presentation.

In the slide Actor IEFSM, it becomes more complicated because of its static nature. The possible situations are as follows. The situation of executing atomic transition AT_3 is the same as that for video and audio streams. The situation of executing atomic transition AT_4 is analyzed as follows. For convenience, variable *rtime* records the remaining display time from an intermediate point to the end point of the current presentation stage under the new presentation flow direction; function *ConvertSlide* calculates the corresponding *scount* and *rtime* according to the value of *index*. The display duration of the new current presentation stage *scount* becomes *rtime* after the reverse action being achieved. To simplify the execution, newly added transitions T_2 and $\overline{T}_2$ are used to display the very first stage after reversing the presentation flow direction, if the slide IEFSM is at state S_2 when the reverse operation is executing. The

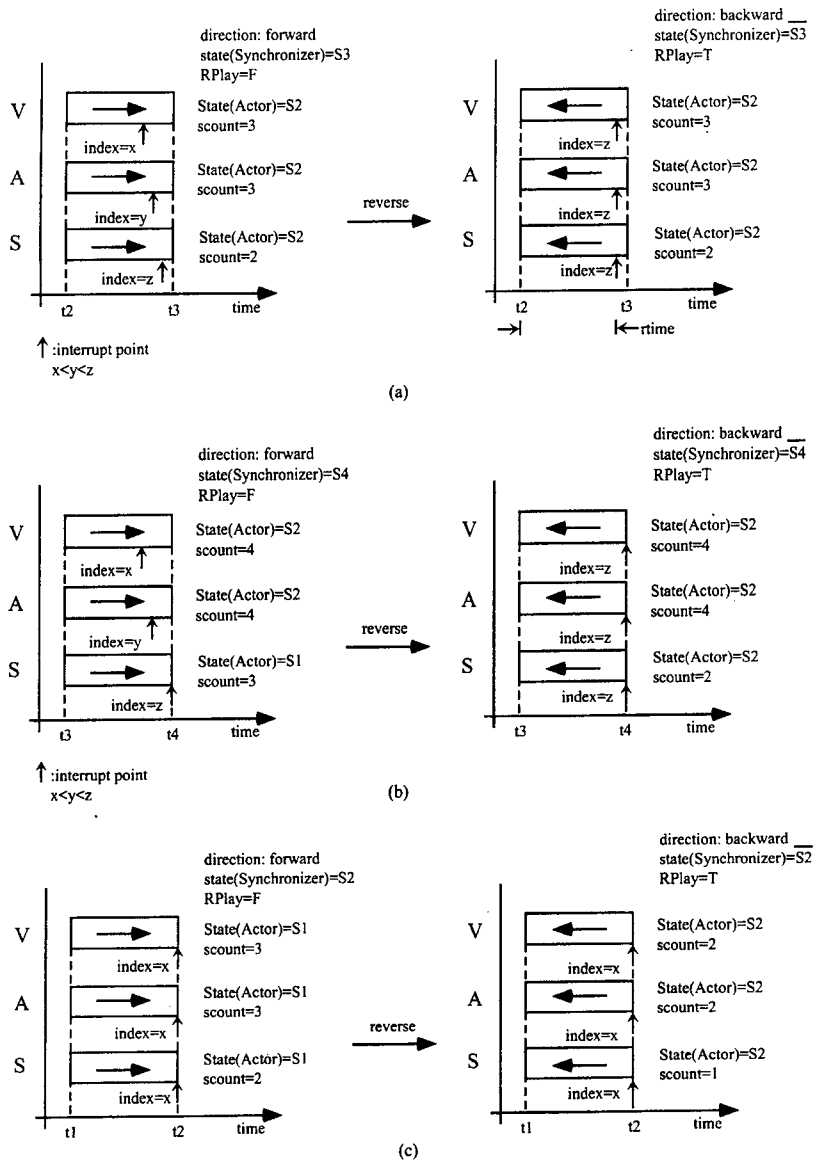


Fig. 12. Some examples of applying the reverse operation, (a) all of these three streams are displaying media units belonging to their current stages and the slide medium is the fastest stream, (b) video and audio streams are displaying media units belonging to the 4^{th} stage, and the slide medium has finished stage 2 and is going to present the media units in stage 3, (c) all of these three streams have just finished their current presentation stages and are going to present their next stage, respectively.

display of the following stages are still in the modified transition T_1 and the newly added transition $\overline{T_1}$. Variable *temp* is used to control whether T_2 ($\overline{T_2}$) needs to be executed or not.

Racing conditions may occur: After an Actor sends the "Finish" ("RFinish") message, the Synchronizer receives a *reverse* message by executing interrupt transition ${}_rIT_i$ (${}_r\overline{IT}_i$) before receiving the "Finish" ("RFinish") message. That is, the Synchronizer may receive messages "Finish" and "RFinish" at states S_i', i=1..6, which are depicted in Fig. 9. Transitions AT'_{ij}, i=1 to 6, and (i) j=1 to 3 if i=2, 4, or 6, and (ii) j=1 to 2 if i=1, 3, or 5, ($\overline{AT'}_{ij}$, i=1 to 6, and (i) j=1 to 3 if i=1, 3 or 5, and (ii) j=1 to 2 if i=2, 4, or 6), which are depicted in Fig. 9, are used to receive the "Finish" ("RFinish") messages when the racing condition occurs in the forward to backward (backward to forward) situation.

Figure 12 shows some examples of applying the reverse operation in different situations.

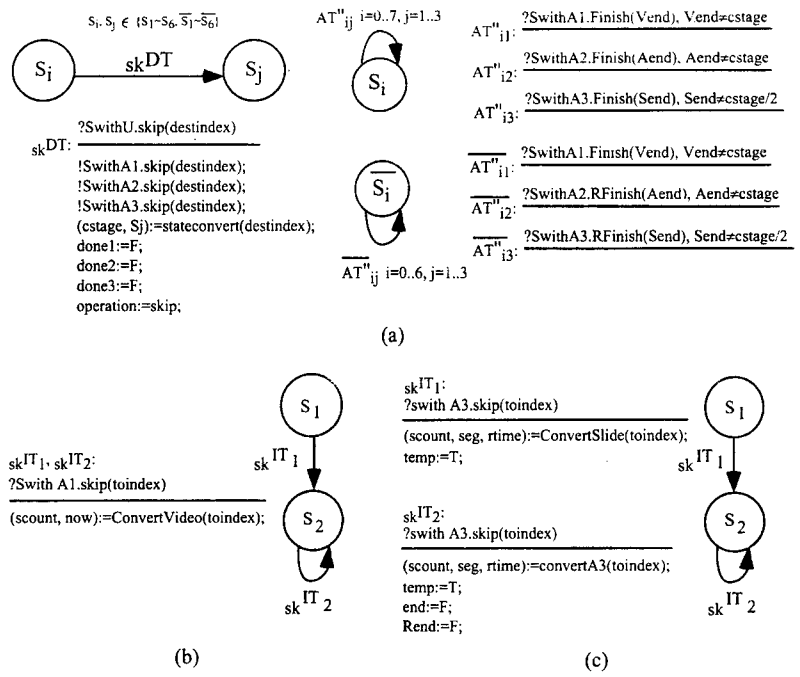


Fig. 13. Modifications in (a) Synchronizer IEFSM, (b) video Actor IEFSM, and (c) slide Actor IEFSM, for the skip operation.

2.The skip operation

The purpose of the skip operation is to allow users to skip from presentation point X to presentation point Y, either at the forward or backward presentation situation. Using skip operations, users are allowed to specify new starting points of presentations. The new starting point can be before or after the interrupt point. The destination of a skip operation is specified in a parameter of the input message.

The modified Synchronizer IEFSM and Actor IEFSMs for the skip operation are depicted in Fig. 13. In the Synchronizer IEFSM, which is depicted in Fig. 13-(a), when a skip message comes in from the user channel, dynamic transition, $_{sk}DT$, is executed. Because the destination of a skip operation can be in any one of the presentation stages, a skip operation may cause state change in the Synchronizer IEFSM if the destination is not in the current presentation stage. *DTs* are atomic transitions and are used to handle dynamic state change in Synchronizer IEFSM. In transition $_{sk}DT$, the destination index is specified in *destindex* of the input message *skip*, and three "skip" messages are sent to the corresponding Actor IEFSMs. Procedure *stateconvert* in $_{sk}DT$ is used to derive the destination state according to the *destindex* parameter.

Racing conditions may occur: If the user issues a skip operation just at the end of a stage, and the "Finish" ("RFinish") messages for this stage (i) have already been sent to Synchronizer, but (ii) have not been received by the Synchronizer, i.e., the Synchronizer receives the "skip" message before receiving all of the "Finish" ("RFinish") messages. The redundant "Finish" messages may exist at any (destination) states, and must be absorbed without conflicting the regular "Finish" ("RFinish") message received at that state. Transitions AT''_{ij}, i=0 to 7, j=1 to 3 ($\overline{AT''}_{ij}$, i=0 to 6, j=1 to 3) are used to receive the redundant "Finish" ("RFinish") messages. To resolve the conflict with the regular "Finish" and "RFinish" messages, predicates "*Vend* ≠ *cstage*", "*Aend* ≠ *cstage*", and "*Send* ≠ *cstage*/2" are used in the corresponding transitions respectively, where *Vend*, *Aend*, and *Send* denote the stage to which the "Finish" ("RFinish") message belongs respectively.

Figures 13-(b) and 13-(c) depict video and slide Actor IEFSMs for the skip operation. For video and audio streams, when a skip message is received by the corresponding Actor IEFSM, one of the interrupt transitions, $_{sk}IT_1$ or $_{sk}IT_2$, which are depicted in Figs. 13-(b) and 13-(c), is executed, depending on whether the state of the IEFSM is at S_1 or S_2 when the skip message is received. In Fig. 13-(b), procedure *ConverVideo*, which is *ConvertAudio* in the audio IEFSM, is invoked in the action parts of interrupt transitions. Procedure *ConvertVideo* derives the corresponding values of variables *scount* and *now*

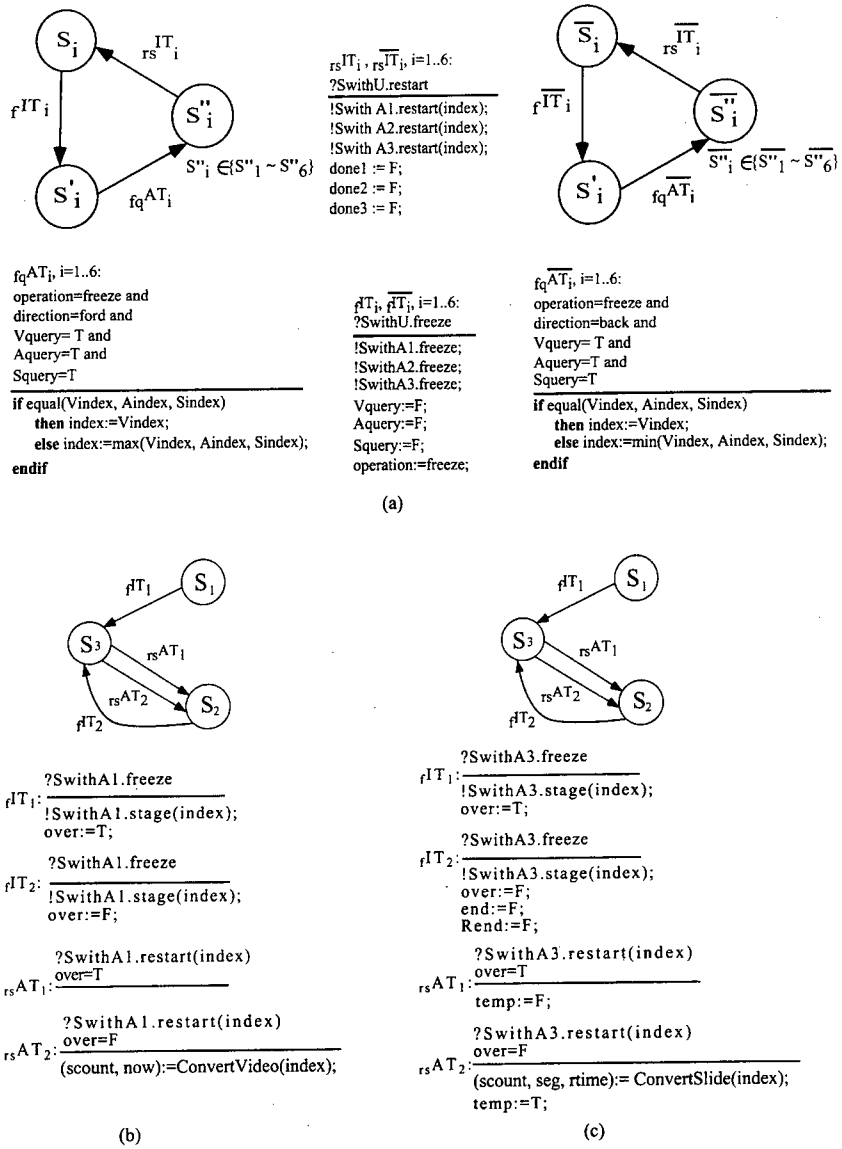


Fig. 14. Modifications in (a) Synchronizer IEFSM, (B) video actor IEFSM, and (c) slide Actor IEFSM, for the freeze-restart operation.

according to the flow index specified in *toindex*. In the interrupt transitions of the slide Actor IEFSM, i.e., $_{sk}IT_1$ and $_{sk}IT_2$ that are depicted in Fig. 13-(c), *toindex* is converted into the destination stage *scount* and the remaining display duration of the new stage *scount*. The destination point of a skip operation can be in the middle of a stage, and if so, only the remaining duration has to be presented. That is, if the destination is in the middle of a stage, the very first presentation stage after the skip operation being executed will be displayed for only part of its full duration. After the very first stage, the Actor displays other stages for their full durations. This requirement is controlled by the *temp* flag.

3. The freeze-restart operation

The purpose of the freeze-restart operation is to allow a user to pause a presentation for a while and then continue the presentation. If the user issues a *freeze* operation, the current presentation of all media streams must be pre-emptied and the resumption of their execution is deferred until the *restart* operation is issued by the user. Fig. 14 depicts the modified Synchronizer IEFSM and Actor IEFSMs for the freeze-restart operation. The interaction sequence for

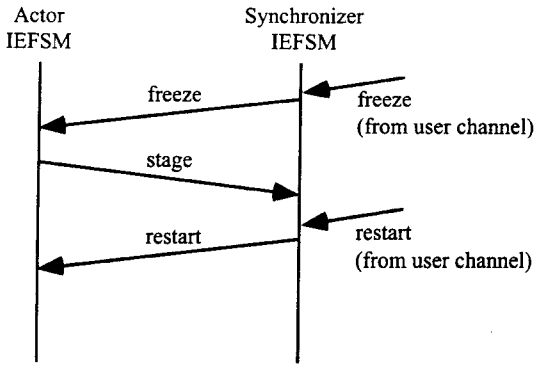


Fig. 15. The interaction sequence for the freeze-restart operation.

the freeze-restart operation is depicted in Fig. 15. After the freeze-restart operation is processed, three media streams must synchronize with each other.

For the Synchronizer IEFSM and the Actor IEFSMs depicted in Fig. 14, video, audio, and slide streams must synchronize with each other when the presentation is resumed after the restart operation has been executed. The re-synchronization policy is similar to that adopted in the reverse operation, i.e., slower streams skip some media units or display time to keep pace with the fastest stream. Since re-synchronization can be achieved by comparing the current flow indices in these three streams, a query processing of current flow indices is required. Two added supplement sets of states, i.e., S''_i and $\overline{S''}_i$, i=1..6, are added. When the "freeze" messages are received by Actor IEFSMs, Actor IEFSMs send *stage* messages, which contain their current flow indices respectively, to the Synchronizer IEFSM. In the Synchronizer IEFSM, after transitions ${}_qAT_{ij}$, which are depicted in Fig. 9, have been executed, transitions ${}_{fq}AT_i$ (${}_{fq}\overline{AT}_i$), which are depicted in Fig. 14-(a), can be executed. The function of transitions ${}_{fq}AT_i$ (${}_{fq}\overline{AT}_i$) is similar to that of transition ${}_{fq}AT_i$ (${}_{fq}\overline{AT}_i$) for the reverse operation, which are depicted in Fig. 9. That is, ${}_{fq}AT_i$ (${}_{fq}\overline{AT}_i$) decides whether the presentation is synchronous or not when the freeze operation is invoked.

Figures 14-(b) and 14-(c) depict the corresponding video and slide Actor IEFSMs respectively. The "Freeze" message can be received at state S_1 or S_2, and variable *over* is set as TRUE or FALSE accordingly. Since slower streams skip some media units or display time to keep pace with the fastest stream, the presentation can be resumed very simply when *over* is equal to TRUE, i.e., transition ${}_{rs}AT_1$ is executed, without invoking additional computation to resume the following presentation. When *over* is equal to FALSE, i.e., the current presentation stage of Actor IEFSM is not finished when the "freeze" message is received, the following presentation flow index after the "restart" message having been received should be calculated, because some asynchrony anomalies may exist among these three streams. Thus, procedure *ConvertVideo*, *ConvertAudio*, and *ConvertSlide* are invoked to derive the new values of *scount* and *now* in video and audio streams, and the remaining display duration in the very first stage of the slide stream. Racing conditions still may exist. Racing conditions that exist in the freeze-restart operation are the same as that for the reverse operation, which are described in Section 6.1.

Some examples of applying the freeze-restart operation are depicted in Fig. 16.

4. The scale operation

The purpose of the scale operation is to allow users to adjust the presentation speed, either faster or slower. Using the scale operation, a user can adjust the presentation speed by some factor to have the presentation much faster or slower. If the factor is greater (less) than one, the presentation will be faster (slower). Fig. 17 depicts the modified Synchronizer IEFSM and the Actor IEFSMs for the scale operation.

For continuous media streams, such as video and audio, faster presentations can be achieved by periodically skipping some frames or segments. For example, if the input factor is 2, the Actor displays one frame for every two frames, e.g., display frames 1, 3, 5, ..., 2k-1, and skip frames 2, 4, 6, ..., 2k. Thus the presentation speed is doubled. On the other hand, if the factor is 0.5, the Actor displays each frame twice, and thus the presentation speed becomes half of the original one. This requirement for skipping or repeating frames is achieved by procedure *Adjust* in transitions ${}_{sc}AT_1$ and ${}_{sc}AT_2$ that are depicted in Fig. 17-(b). For static media streams, such as slide, the duration of every stage and the remaining time of the current stage is multiplied with the scaling factor, which is achieved in transitions ${}_{sc}AT_1$ and ${}_{sc}AT_2$ that are depicted in Fig. 17-(c).

Figure 17 depicts the corresponding Synchronizer IEFSM and Actor IEFSMs for the scale operation. Since the corresponding actions are the same as that in executing reverse and freeze-restart operations, the explanation and example are dropped for simplicity.

VII. CONCLUSION AND FUTURE WORK

The demand for multimedia systems is increasing. Synchronization of multiple streams with user interactions has been recognized as one of the significant and key issues for having successful multimedia applications [3]. In this way, users are

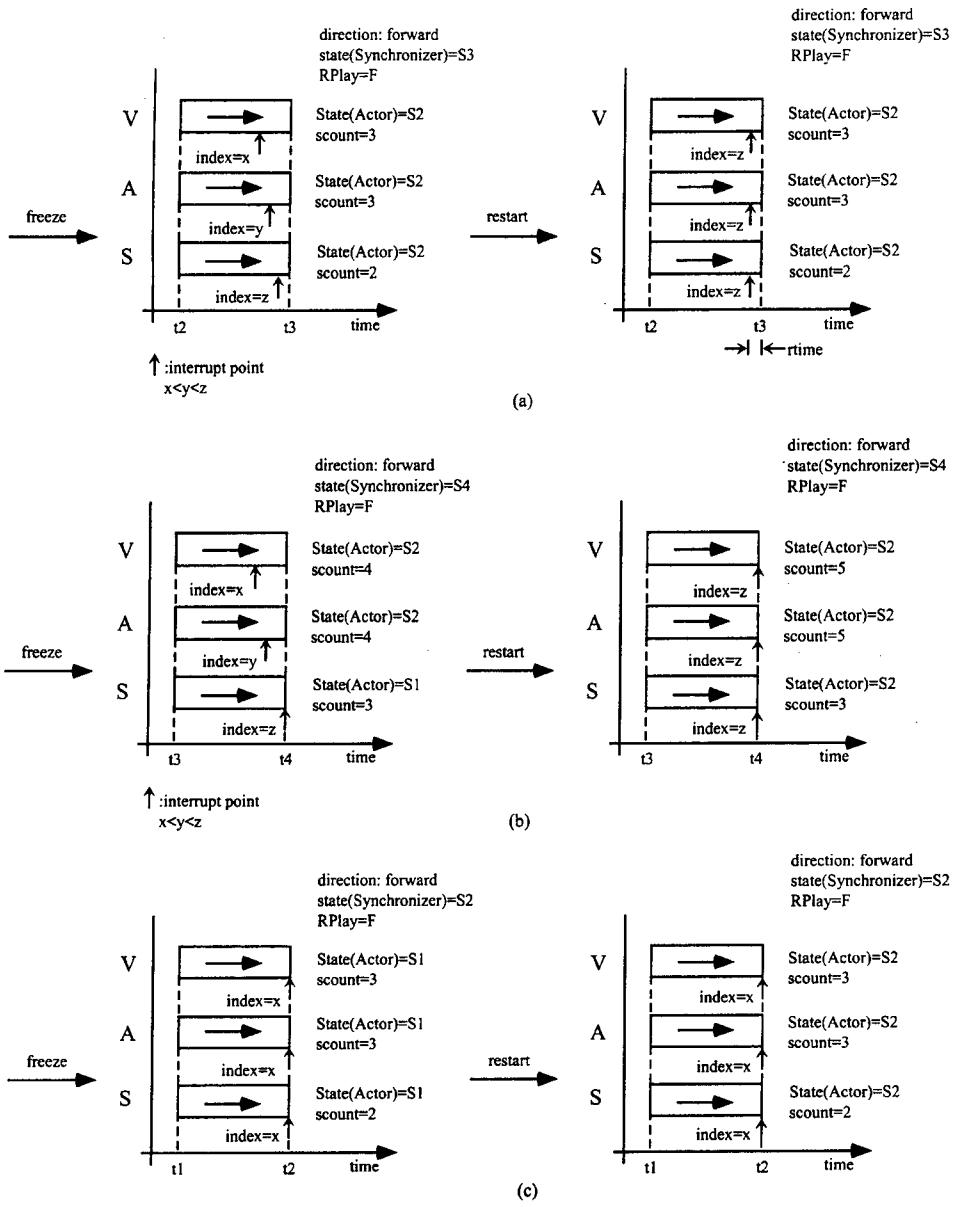


Fig. 16. Some examples of applying the freeze-restart operation, (a) all of these three streams are displaying media units belonging to their current stages and the slide medium is the fastest stream, (b) video and audio streams are displaying media units belonging to the 4^{th} stage, and the slide medium has finished stage 2 and is going to present the media units in stage 3, (c) all of these three streams have just finished their current presentation stages and are going to present their next stage, respectively.

allowed to manipulate display sequences at any time during multimedia presentations. In this paper, we have proposed an IEFSM model that is able to model a synchronous multimedia presentation with user interactions. Both inter-media and intra-medium synchronization issues are considered and formally specified using the IEFSM model. Using the IEFSM model, inter-media synchronization is handled by a Synchronizer IEFSM, and intra-medium synchronization is handled by some Actor IEFSMs. User interactions, such as reverse, skip, freeze-restart, and scale are resolved by using interrupt transitions and dynamic transitions in the IEFSM model. The proposed IEFSM model has the following characteristics:

- The behavior of each medium stream can be

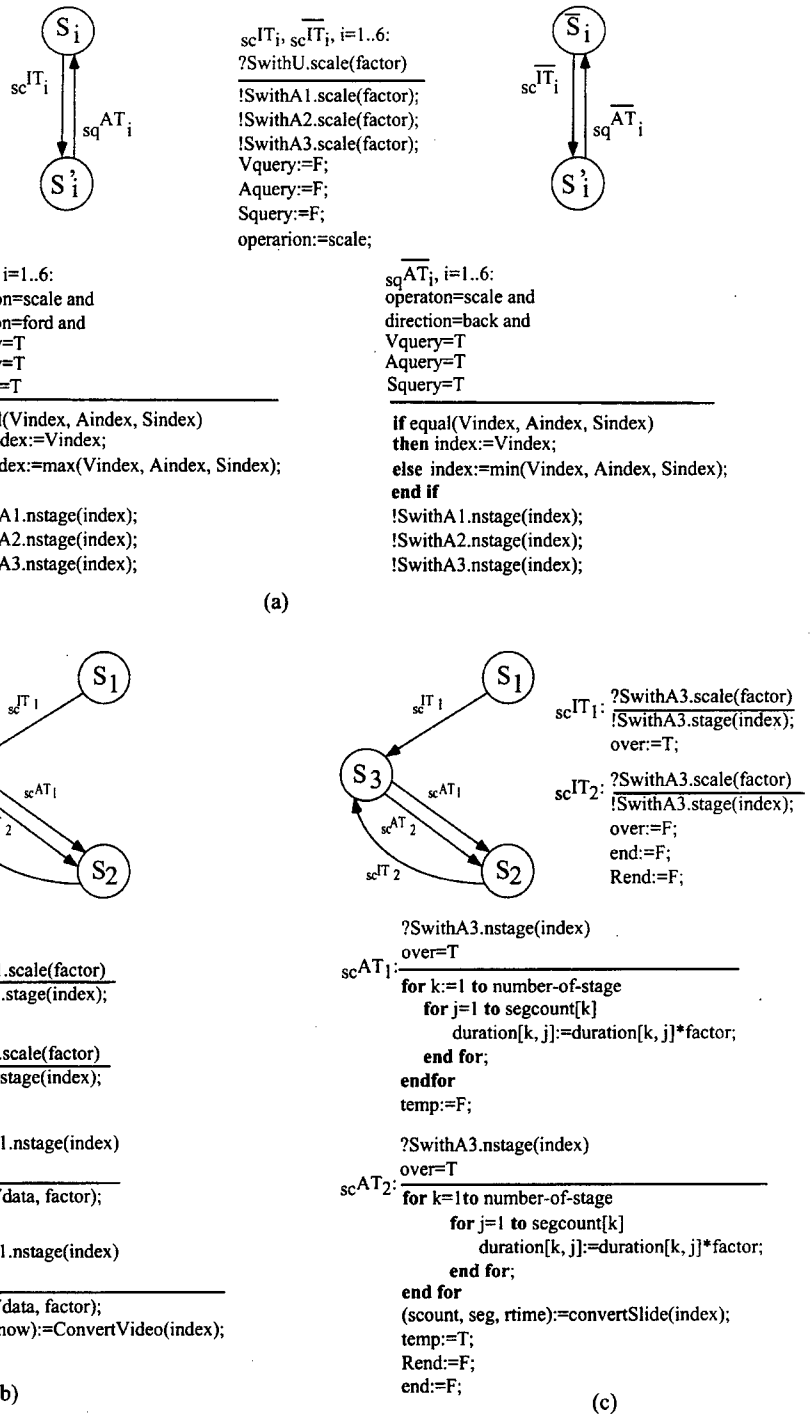


Fig. 17. Modifications in (a) Synchronizer IEFSM, (b) video Actor IEFSM, and (c) slide Actor IEFSM, for the scale operation.

specified.

- The dynamic configurations of inter-media synchronization and intra-medium synchronization with user interactions can be specified.
- Re-synchronization policies, which are needed when asynchrony anomalies exist and the resumed

presentations should be continued synchronously, can also be specified.

We are currently defining an IEFSM-based language and the corresponding language compiler. In this way, the corresponding specification language and compiler can be adopted to multimedia software that need interactive features, e.g., Video-On-Demand (VOD), News-On-Demand (NOD), and distant learning, as the synchronization controllers in the embedded systems.

ACKNOWLEDGMENT

The research is supported by the National Science Council of the Republic of China under the grant NSC 86-2213-E006-071.

REFERENCES

1. Allen, J.F., "Maintaining Knowledge about Temporal Intervals," *Communications of the ACM*, Vol. 26, No. 11, pp. 832-843 (1983).
2. Anderson, D.P. and G., Homsy, "A Continuous Media I/O Server and Its Synchronization Mechanism," *IEEE Computer*, Vol. 32, No. 5, pp. 51-57 (1991).
3. Blakowski, G. and R., Steinmetz, "A Media Synchronization Survey: Reference Model, Specification, and Case Studies," *IEEE Journal on Selected Areas in Communications*, Vol. 14, No. 1, pp. 5-35 (1996).
4. Brand, D. and P., Zafiropulo, "On Communicating Finite State Machines," *Journal of ACM*, Vol. 30, No. 2, pp. 323-342 (1983).
5. Buchanan, M.C. and P.T., Zellweger, "Automatically Generating Consistent Schedules for Multimedia Applications," *ACM Multimedia Systems*, Vol. 1, No. 2, pp. 55-67 (1993).
6. Chen, H.Y. and J.L., Wu, "MultiSync: A Synchronization Model for Multimedia Systems," *IEEE Journal on Selected Areas in Communications*, Vol. 14, No. 1, pp. 238-248 (1996).
7. Eun, S. No, E.S. Kin, H.C. Yoon, H.S. and Maeng, S.R., "Eventor: an Authoring System for Interactive Multimedia Applications," *ACM Multimedia Systems*, No. 99, pp. 129-140 (1993).
8. Furht, B., "Multimedia Systems: An Overview," *IEEE Multimedia*, Vol. 1, No. 1, pp. 47-59 (1994).
9. Grosky, W.I., "Multimedia Information Systems," *IEEE Multimedia*, Vol. 1, No. 1, pp. 12-24 (1994).
10. Haindl, M., "A New Multimedia Synchronization Model," *IEEE Journal on Selected Areas in Communications*, Vol. 14, No. 1, pp. 73-83 (1996).
11. Hirzalla, N.B. Falchuk, and A., Karmouch, "A Temporal Model for Interactive Multimedia Scenarios," *IEEE Multimedia*, Vol. 2, NO. 3, pp. 24-31 (1995).
12. Huang, C.M. and C.M., Lo, "An EFSM-based Multimedia Synchronization Model and the Authoring System," *IEEE Journal on Selected Areas in Communications*, Vol. 14, No. 1, pp. 138-152, January (1996).
13. Lamont, L. and N.D., Georganas, "Synchronization Architecture and Protocols for a Multimedia News Service Application," Proc. of *the 1st IEEE International Conference on Multimedia Computing and Systems*, pp. 3-8 (1994).
14. Li, L. Karmouch, A. and N.D., Georganas, "Multimedia Teleorchestra with Independent Sources: Part 1 - Temporal Modeling of Collaborative Multimedia Scenarios," *ACM Multimedia Systems*, Vol. 1, No. 4, pp. 143-153 (1994).
15. Little, T.D.C. and A., Ghafoor, "Synchronization and Storage Models for Multimedia Objects," *IEEE Journal on Selected Areas in Communications*, Vol. 8, No. 3, pp. 413-427 (1990).
16. Prabhakaran, B. and S.V., Raghavan, "Synchronization Models for Multimedia Presentation with User Participation," *ACM Multimedia Systems*, Vol. 2, pp. 53-62 (1994).
17. Schnepf, J. J.A. Konstan, and D.H.C., Du, "Doing FLIPS: Flexible Interactive Presentation Synchronization," *IEEE Journal on Selected Areas in Communications*, Vol. 14, No. 1, pp. 114-125 (1996).
18. Steinmetz, R., "Synchronization Properties in Multimedia Systems," *IEEE Journal on Selected Areas in Communications*, Vol. 8, No.3, pp. 401-412 (1990).
19. Vazirgiannis, M. and C., Mourlas, "An Object-Oriented Model for Interactive Multimedia Presentations," *The Computer Journal*, Vol. 36, No. 1, pp. 78-86 (1993).
20. Yang, C.C. and J.H., Huang, "A Multimedia Synchronization Model and Its Implementation in Transport Protocols," *IEEE Journal on Selected Areas in Communications*, Vol. 14, No. 1, pp. 212-225, January (1996).
21. Woo, M.N. Qazi, and A., Ghafoor, "A Synchronization Framework for Communication of Preorchestrated Multimedia Information," *IEEE Network*, Vol. 8, No. 1, pp. 52-61 (1994).

APPENDIX

Comparison between the FSM-based and the IEFSM-based specifications

Figures18 and 19 depict the Synchronizer and Actor FSMs for the illustrated presentation, which is depicted in Fig. 5, respectively. The main difference between IEFSMs and FSMs is that the condition part

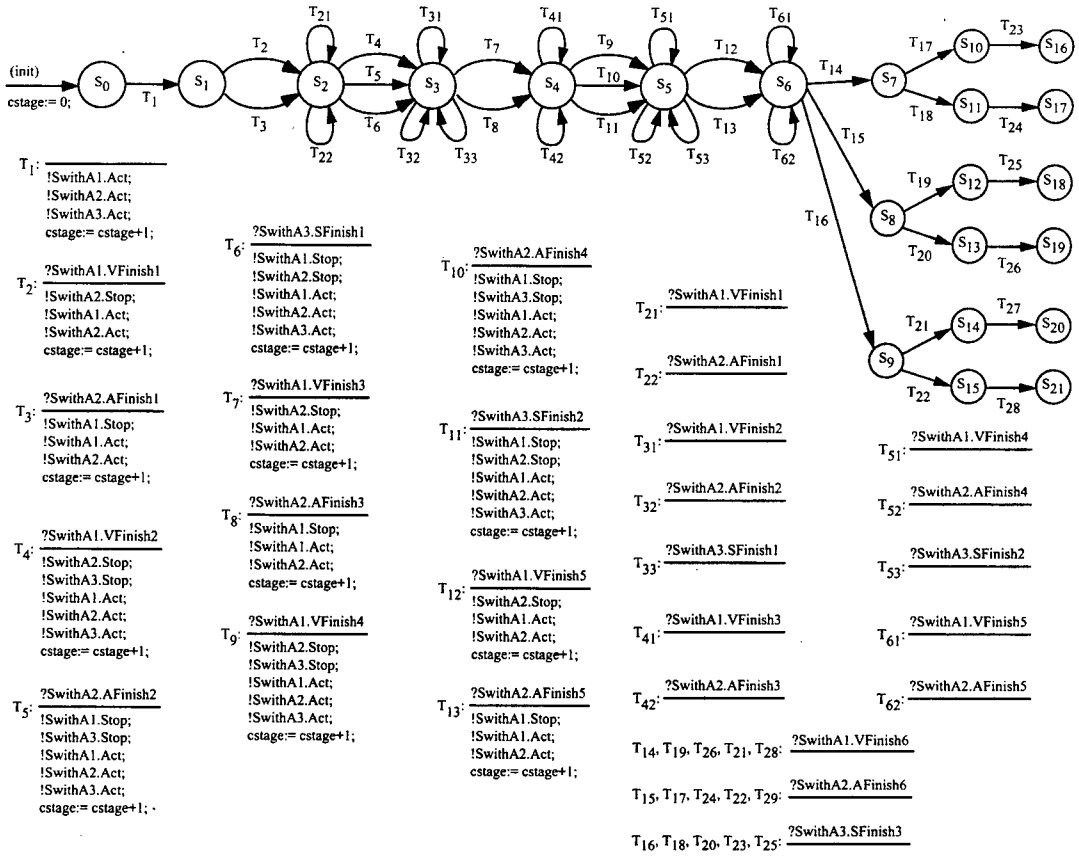


Fig. 18. The Synchronizer FSM for the presentation depicted in Fig. 5.

of an FSM's transition contains only input event(s) without any predicate. Figure 18 depicts a Synchronizer FSM that adopts the parallel-first inter-media synchronization policy [12]. When the Synchronizer receives a *Finish* message that is sent from the video, audio, or slide Actor, the Synchronizer sends *Stop* messages to other Actors to stop the presentation of the current stage. In the IEFSM model, only one transition is enough to specify the activation of the next presentation stage; but in the FSM model, n transitions, e.g., transitions T_4, T_5, and T_6 in Fig. 18, are needed to specify the activation of the next presentation stage, where n is the number of Actors. For the last presentation stage, all Actors have to finish their display and then the presentation can be stopped. Therefore, the parallel-last inter-media synchronization policy is adopted for the last presentation stage [12]. In the IEFSM model, only one transition is enough to specify the end presentation stage; however, in the FSM model, $n!$ transitions are needed to specify the end presentation stage, where n is the number of Actors. For example, since there are three Actors in Fig. 18, there are 3!=6 different sequences for the Synchronizer's receiving *Finish* messages. Additionally, it is possible that the Synchronizer sends a *Stop* message and an Actor sends a *Finish* message concurrently. In order to avoid this racing situation, i.e., the Synchronizer receives a *Finish* message after it has sent *Stop* messages, some transitions are needed to absorb the redundant *Finish* messages that are for the previous stage. Transitions T_{21}, T_{31}, T_{41}, T_{51}, and T_{61} (T_{22}, T_{32}, T_{42}, T_{52}, and T_{62}) are used to absorb the video (audio) *Finish* message that is for stages 1, 2, 3, 4, and 5 respectively. Transitions T_{33} and T_{53} are used to absorb the slide *Finish* message that is for stages 1 and 2 respectively.

Figures 19-(a) and 19-(b) depict the video and slide FSMs for the illustrated presentation respectively. Because the audio FSM is similar to the video FSM, the audio FSM is not depicted for simplicity. In Fig. 19, only FSMs for the first stage are depicted. FSMs for other stages can be derived accordingly. If there are $no[i]$ video frames in stage i, then there are $no[i]$ states and transitions for displaying these video frames, i.e., states S_{11} to $S_{1no[scount]}$ and transitions T_{11} to $T_{1no[scount]}$. State S'_1 and transitions T'_{11} to

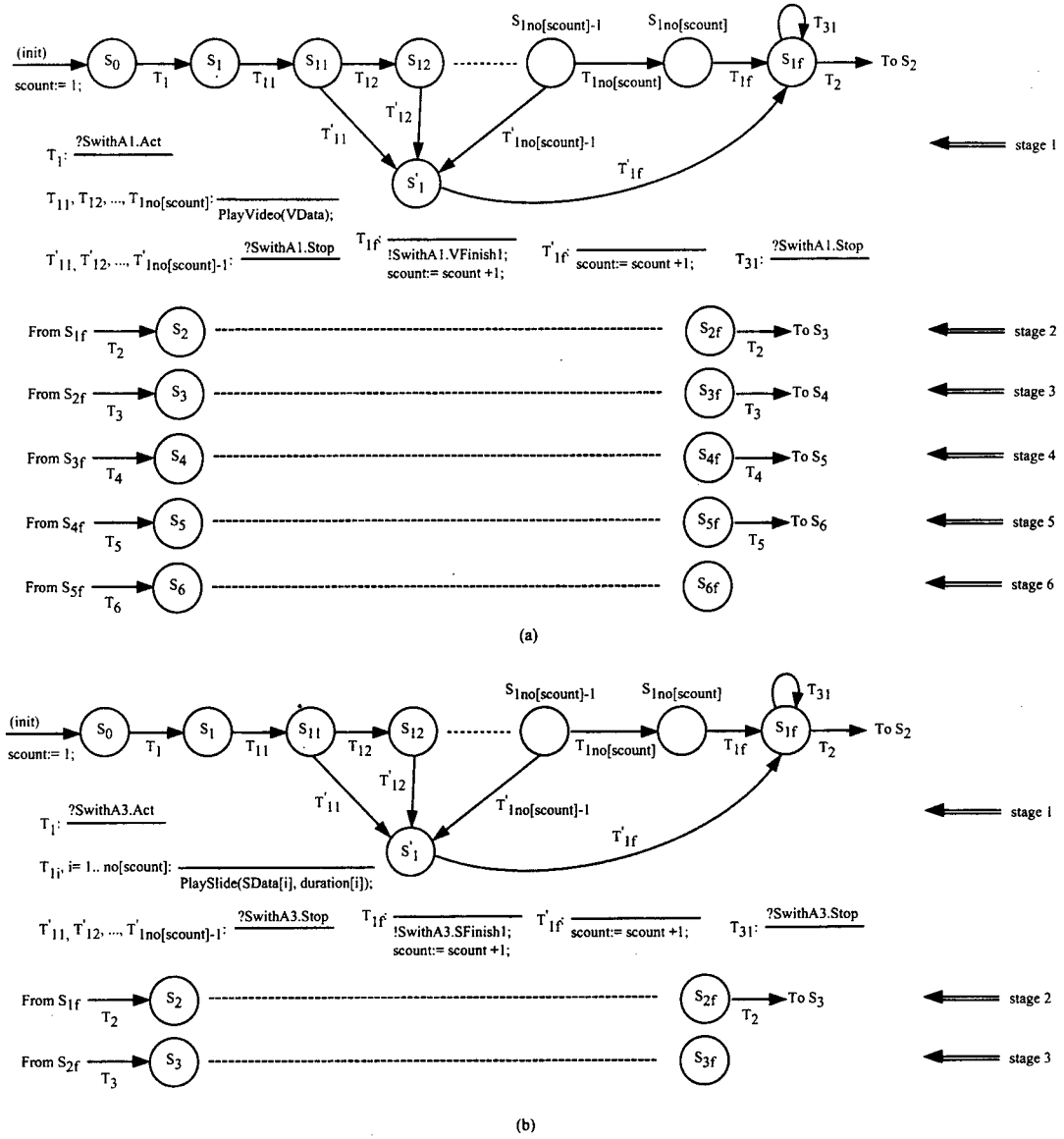


Fig. 19. (a) The video and (b) the slide Actor FSMs for the presentation depicted in Fig. 5.

$T'_{1no[scount]-1}$ are for processing the *Stop* message. Similarly, for the slide stream, if there are $no[i]$ slides in stage i, then there are $no[i]$ states and transitions for displaying these slides, i.e., states S_{11} to $S_{1no[scount]}$ and transitions T_{11} to $T_{1no[scount]}$. State S'_1 and transitions T'_1 to $T'_{1no[scount]-1}$ are for processing the *Stop* message. Compared with the IEFSMs that are depicted in Fig. 6, there are many more states and transitions in the FSMs, which are depicted in Figs. 18 and 19.

Discussions of this paper may appear in the discussion section of a future issue. All discussions should be submitted to the Editor-in-Chief.

Manuscript Received: Aug. 27, 1996
Revision Received: July 20,1997
and Accepted: Jan. 20, 1998

以 Interactive Extended Finite State Machines (IEFSMs) 爲基礎的多媒體互動同步機制

黃崇明　王謙

國立成功大學資訊工程研究所

摘　要

現今多媒體系統的特性之一便是互動功能的提供。對隨選視訊(Video-On-Demand, VOD)和隨選新聞(News-On-Demand, NOD)來說，互動功能更是必需的功能之一。互動功能的提供，讓使用者在觀賞或瀏覽時更具彈性。也就是說，由於互動功能的提供，使用者可以即時地控制多媒體資訊的播放流程，例如：快轉、倒放等。在這一篇論文中，我們提出了Interactive Extended Finite State Machines (IEFSMs)模式來描述並解決互動式多媒體系統上的同步問題。在我們所提出的 IEFSM 模式中，單一媒體內的同步 (intra-medium synchronization) 是由 Actor IEFSM 來控制，而不同媒體間的同步 (inter-media synchronization) 是由 synchronizer IEFSM 來控制。藉由 Synchronizer 及Actor的溝通與合作，互動功能的動態性及不確定性便可以被加以描述，並進而解決所發生的同步問題。

關鍵詞：多媒體，同步，擴充性有限狀態機，互動功能。

WAVEFORM APPROXIMATION TECHNIQUE FOR CMOS GATES IN THE SWITCH-LEVEL TIMING SIMULATOR BTS

Molin Chang, Jyh-Herng Wang, Shuih-Jong Yih and Wu-Shiung Feng*
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan 107, R.O.C.

Key Words: waveform approximation technique, switch-level timing simulation, RC tree.

ABSTRACT

A switch-level timing simulator has the advantage of fast speed and good adaptability for VLSI circuits, but it cannot offer accurate transient waveform information. In this paper an accurate and efficient switch-level timing simulator is described. The high accuracy is attributed to a new waveform approximation technique, which includes delay estimation and slope estimation. Efficient delay and slope calculations are accomplished through a switch-level simulation instead of using a transistor-level simulation. A new approach for delay estimation is presented which models the delay behavior of an RC tree by two equations: a dominant delay equation and an error delay equation. Both are derived by surface fitting to approximate the surface that is measured from the actual delay behavior of a CMOS gate. A modified approach for slope estimation is also investigated which has close relationship with the equivalent RC time constant of the evaluated cluster circuit. This equivalent RC time constant can be obtained by traversing the tree recursively. The results show good agreement with SPICE.

I. INTRODUCTION

The simulation of MOS logic circuits is an important process before fabrication because of the high cost of manufacturing an integrated circuit, especially the very large scale integrated (VLSI) circuits. Many types of simulations can be applied to this work. In general, we have two extreme approaches to simulate a circuit; one is taken by the circuit-level simulator, such as SPICE, and the other is taken by the logic-level simulator. Circuit Simulators are almost always used to simulate portions of an IC or small circuits at the transistor level. They are extremely accurate and flexible but they also require expensive computation time and a large amount of memory. On the other hand, logic simulators are much faster than the circuit simulator and can be used for large circuits, but they perform less accurate transient analysis.

Because of this, switch level simulation, which is a compromised method, has been proposed and some switch level simulators have also been implemented, such as MOSSIM, RSIM [1], [10]. We have also developed a switch level simulator: Binary tree Timing Simulator (BTS), which is three orders faster than SPICE and has more accurate waveform approximation during the transient state.

Most switch-level algorithms emphasize how to

*Correspondence addressee

calculate the time constant of charging/discharging the load capacitance more accurately. There is much research on this topic. Penfield et al. [9] presented bounds with a fixed level of accuracy for the delay in RC trees. The Elmore delay [7], in the RC tree in particular, can be computed extremely efficiently, as was shown by Lin and Mead with their algorithm TREE [8], although this is only a reasonably good approximation to the true delay. For more general networks, many methods were proposed in the past. Lin and Mead [8] proposed an iterative algorithm which was based on converting the nontree-like RC networks to an RC tree using node splitting. Caisso et al. [2] proposed an efficient algorithm for computing the Elmore delay in RC networks in which the resistors are interconnected in a series-parallel manner. However, all of above cannot offer us more accurate waveform information in the transient state; we want to know not only whether the logic gate changes state or not, but also when the output voltage begins to change and how fast it will change.

The high accuracy of BTS is attributed to a new waveform approximation technique, which includes delay estimation and slope estimation. The *delay* estimation tells us when the output begins to change and the *slope* estimation tells us how fast the output will change. First, we discuss the delay estimation. An uncertain amount of overshoot, chiefly due to parasitic capacitors, will almost always be produced at the output node while an event is happening at the input. The width of overshoot is the keypoint; if it can be predicted well, and then the delay will be estimated accurately. Second, the slope relates closely to the RC time constant of the discharging/charging path; it is only a constant relation. Lin and Mead proposed an efficient method that can be implemented in a recursive way. Furthermore, another important feature of BTS is that the delay and slope calculations are considered with internal charges and charge sharing effects. The internal charges stored in the internal nodes of a MOS circuit will increase the delay time about 20% when the tested circuit is a five-input NAND gate with four fully charged internal nodes. Therefore, the effect of internal charges should also be considered when the delay time is estimated. Besides that, we should consider the effect of internal charges in a *non-active tree*, which is stated in section IV, when we estimate the slope. For example, if there is a falling signal on the output, the internal nodes connected to the output node in the N tree (*active tree*) are considered when calculating the delay, and the internal nodes connected to output in the P tree (*non-active tree*) should be taken into account when calculating the slope, except that the RC time constant of the N tree is computed first.

In this paper, we describe the concept of BTS in section II. Next, the waveform approximation technique is presented. Active and non-active trees are defined in section IV. Then, in sections V and VI, the delay and slope estimation, respectively are stated in more detail. Finally, the simulation results are shown in section VII.

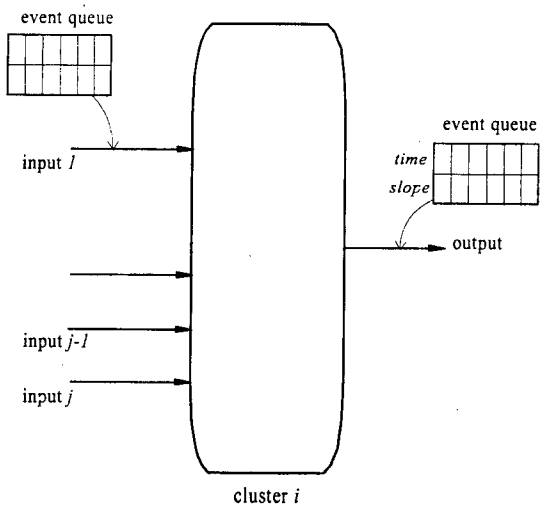


Fig. 1. Evaluation of a cluster.

II. BTS'S TIMING SIMULATION

BTS is an event-driven switch-level timing simulator. The simulator reads in the circuit description file and partitions the circuit into groups of nodes connected by source-drain channels. Such groups are known as clusters (or *blocks* in BTS). An event-driven scheduling algorithm (levelization of clusters) is used to schedule the evaluation of clusters. If there is no feedback existing in the circuit, a waveform-evaluation technique is used to obtain the whole waveform of each cluster, one by one from the primary input to the primary output of the circuit. However, if a feedback path exists in the circuit, the conventional event-driven technique is used, such as event evaluation, event propagation and event insertion and deletion. Because voltage changes on the source and drain cannot affect the gate (i.e., the gate coupling is unidirectional), the evaluation of a cluster can be proceeded independently of all other clusters once its inputs are known. The data structure of each edge contains an event list that records all events happening at this edge. After all events are processed, the whole voltage waveform at each edge can be obtained from the event list (see Fig. 1), which contains two terms, time and slope, to represent each event. BTS uses the waveform approximation technique, described in next section, to reconstruct the waveform.

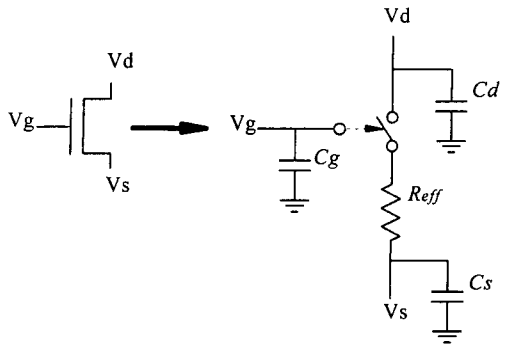


Fig. 2. The MOS model used in BTS.

1. MOS model

The MOS model in BTS is composed of a voltage-controlled switch, effective resistance R_{eff} and equivalent grounded capacitances, as shown in Fig. 2. The transistor is ***on*** (the switch conducts) if and only if the gate voltage of the NMOS transistor is higher than its threshold voltage V_T. The turn-on effective resistor is distinguished by two cases: R_{on} (in a steady state) and R_t (in a transient state), because the MOS transistor (denoted by MOSt) has different response under different gate states. Therefore, the value of R_{eff} may be one of the three cases:

$$R_{eff}=\begin{cases}\infty & \text{if } V_g < V_T \\ R_{on} & \text{if } V_g \text{ is high (steady state)} \\ R_t & \text{if } V_g \text{ changes from L to H (transient state)}\end{cases} \quad (1)$$

The values of R_{on} and R_t depend on the physical parameters and the load capacitance, and R_t also depends on the slope of the signal at the gate. These two values, R_{on} and R_t, can be obtained from the simulated results of the circuit in Fig. 3 by using SPICE. In case I of Fig. 3, V_g is 5V and SW is a voltage-controlled switch. The falling waveform of a simple RC circuit can be represented as

$$V_O = V_{DD}\exp(-\frac{t}{RC}), \quad (2)$$

therefore $T_{10\%\text{-}50\%}$=0.588RC. We measure the duration (t_2'–t_1') from 90% (t_1') to 50% (t_2') of the falling waveform of the output V_o and solve the equations below:

$$0.588R_{on}C_d=T_1 \quad \text{withoutCload} \quad (3)$$

$$0.588R_{on}(C_d+C_{load})=T_1 \quad \text{withCload} \quad (4)$$

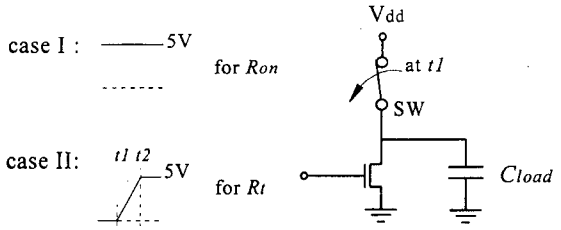


Fig. 3. The tested circuit for obtaining the R_{on} and R_t of a MOS transistor.

where T_1 and T_2 are the difference between t_2' and t_1', and C_d, C_{load} are drain and load capacitances, respectively. In case II of Fig. 3, the concept is the same as mentioned above except that Vg changes from low (at t_1) to high (at t_2) and SW is opened at t_1. Assuming C_d is constant, we can obtain the values of R_t from the following equation with respect to different input slopes.

$$0.588R_tC_d=T \quad (5)$$

where the definition of T is the same as T_1 and T_2.

In the actual implementation, we maintain two tables in our program, which are two-dimensional Ron-table and three-dimensional Rt-table,

$$R_{on}=f(W/L,C_{load}), \text{ and} \quad (6)$$

$$R_t=f(W/L,C_{load},slope). \quad (7)$$

Because the load capacitance of a gate circuit is the summation of the input capacitances of the fanouts of this gate circuit, C_{load} will be discrete times of the input capacitance of a CMOS inverter with the minimum size (W/L). From the simulated results of SPICE, the relationship between the effective resistance and the slope of the gate signal is smoothly linear, so only a few slopes need be recorded, and the actual value can be obtained using interpolation.

2. Series-parallel tree

If a gate circuit (also called a cluster) is connected in a series-parallel manner such as fully complementary CMOS, Pseudo-NMOS, Dynamic CMOS and so on, it can be represented as a merged series-parallel tree. A merged tree, also called a PN tree, consists of two series-parallel trees, which are the left subtree (P tree) and the right subtree (N tree). When it is mapped to a gate circuit, the left subtree and the right subtree represent the pull-up and pull-down subcircuits, respectively. Fig. 4(b) illustrates the corresponding PN tree of the gate circuit as shown

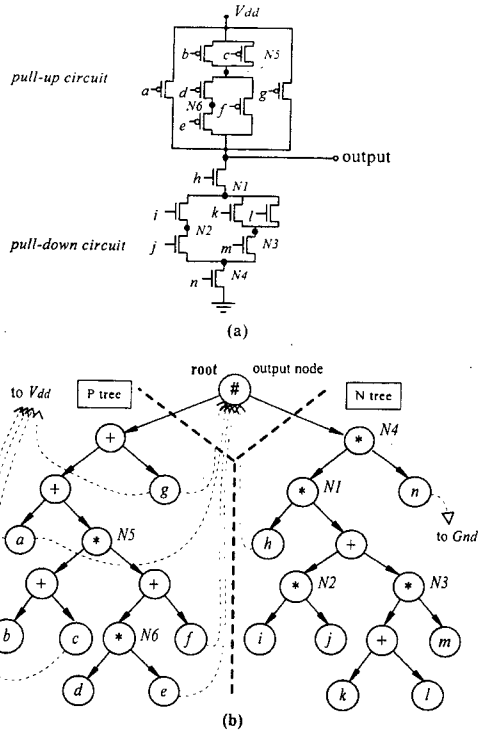


Fig. 4. CMOS complex gate. (a) circuit diagram , (b) the equivalent merged series-parallel tree.

in Fig. 4(a) whose function is

$$Z=(h\bullet((i\bullet j)+((k+l)\bullet m)))\bullet n. \quad (8)$$

In order to represent the non-complementary CMOS circuits, a new function expression is used. It's general form is **Z=(the function of P tree)#(the function of N tree)**, where # represents the root node of a PN tree. For fully complementary CMOS, the function of the P tree is just a complement of that of the N tree. If *a*, *b*, ..., *g* in Fig. 4(a) are replaced with *h*, *i*, ..., *n* then Eq. 8 can be rewritten as follows:

$$Z=((h+((i+j)\bullet((k\bullet l)+m)))+n)\#((h\bullet((i\bullet j)+((k+l)\bullet m)))\bullet n) \quad (9)$$

where • and + represent AND and OR operations, respectively. Note that the inversion of each input in the function of the P tree is absorbed by PMOS.

Figure 4 is the demonstrative example in this paper. Although it is a complementary CMOS circuit, the labels *a*, *b*, ..., *g* are still reserved because of the necessity for demonstrating the non-active tree effect on the slope estimation in section VI.

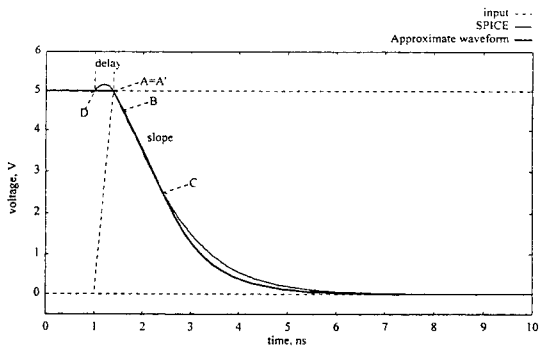


Fig. 5. Waveform approximation by delay and slope estimation method : type A.

The equivalent PN tree of each cluster is established once at the beginning of simulation. The calculations of the equivalent resistances, the equivalent RC time constant and the charge sharing effect of a circuit can be solved efficiently because they are based on the structure of a series-parallel tree [4] [5].

3. Internal nodes and internal charges

Each MOSt in the circuit is a leaf node in the PN tree, and each connection between MOSt's will produce an operator node in the tree. Each operator node has the capacitance summed by all the values of parasitic capacitors on the MOSt's terminals that are connected to this operator node. For example, the operator node N_4 in Fig. 4(a) has the capacitance summed by two C_s's of M_j and M_m and one C_d of M_n, where M_i is the MOSt whose input is named *i*. Charges can be stored in these capacitors and also can be transferred among them while some of inputs are changed. Operator nodes connected to *Vdd* (power), *Gnd* (ground), or C_{load} through the source-drain channel of a MOSt are called the *internal nodes*; thus, the charges stored in internal capacitors that are used to model the drain and source capacitances of MOSt's are called the *internal charges*. In contrast, the operator nodes connected to C_{load} directly are called *external nodes*; then the charge stored in the load capacitor is, of course, called the *external charge*. For example, the operator nodes N_1 to N_6 in Fig. 4(a) are internal nodes and the output node is an external node. The internal nodes play an important role on the charging/discharging behavior of the output node.

III. WAVEFORM APPROXIMATION

The approximation work can be simplified if we cut off the overshoot and use a linear segment

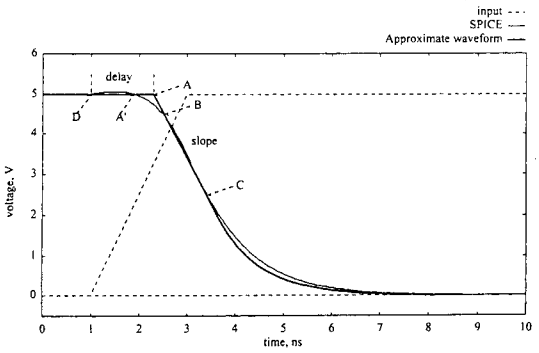


Fig. 6. Waveform approximation by delay and slope estimation method : type B.

followed by an exponential tail to approach the falling (or rising) signal. Fig. 5 illustrates this idea, and the bold solid line will fit the SPICE result well if we can calculate the time of point A and the slope of the linear segment between point B and point C. Next, we use two equations as follows to plot the transient waveform [3].

$$f=\begin{cases}\dfrac{0.2t}{T} & t<3T \\ 1-0.4\exp(-\dfrac{t-3T}{2T}) & t\geq 3T\end{cases} \quad \text{for a rising signal} \tag{10}$$

$$f=\begin{cases}1-\dfrac{0.2t}{T} & t<3T \\ 0.4\exp(-\dfrac{t-3T}{2T}) & t\geq 3T\end{cases} \quad \text{for a falling signal} \tag{11}$$

where T is half of the time spent by the signal between 90% (for a falling signal) or 10% (for a rising signal) and 50% of the steady state. If the value of T can be obtained, the transient waveform will then be easily plotted. Hereafter, we only take the falling signal as an example.

In general, there are two types of waveforms; one is type A as shown in Fig. 5, which has a more near linear segment between point A' and point C, and the other is type B as shown in Fig. 6, which has a larger curvature of the curve between both points A' and C. However, for both types we use a linear segment to fit them. Thus, in contrast to the type A whose delay time is almost equal to the width of the overshoot, the delay time (t_A-t_D) of type B is larger than the actual width ($t_{A'}$-t_D) of the overshoot. Because of this, the definition of delay and slope is not based on the actual waveform, but on the approximate waveform.

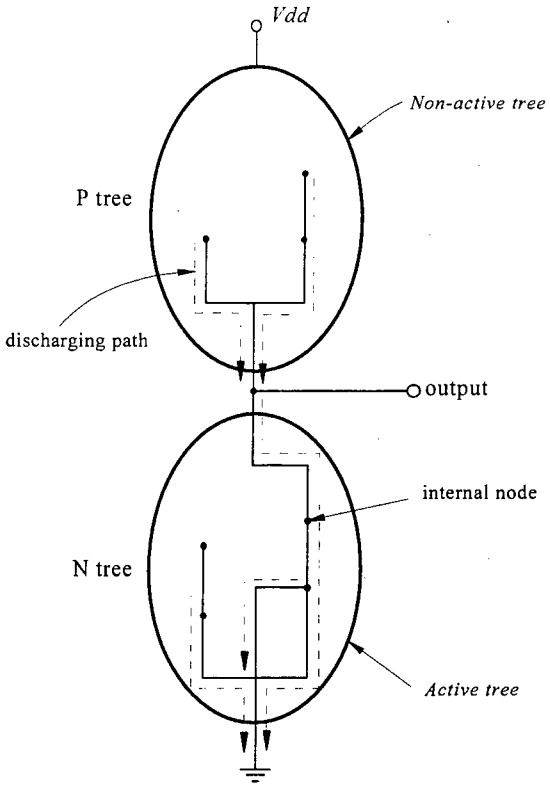


Fig. 7. Active tree and Non-active tree.

Definition1:

The difference between the time when the output signal begins to change and the time when the input signal begins to change is defined as *switching delay* or *delay*, which is denoted by **D**.

Definition2:

The changing rate after the output begins to change, but is restricted between 90% and 50% of the steady state for a falling signal, is defined as *slope*, which is denoted by **S**.

In Fig. 6 the time between point D and point A is the *delay*, and the changing rate between point B and point C is the *slope*. Choose some sample circuits and measure the values for delay time of them from the waveforms resulting from the SPICE simulation. Next, model the behavior of delay by several dominant factors such as input slope, Cl_{oad}, and so on. After the model is established, the delay prediction of an arbitrary circuit is possible.

IV. ACTIVE AND NON-ACTIVE TREE

The PN tree can be illustrated by two blocks as shown in Fig. 7. If the P tree or N tree can let the output node be connected to the source, i.e. V_{dd} (for

the P tree) or *Gnd* (for the N tree), through the paths formed by the drain-source channels of the turn-on MOSt's, then this tree is called an ***active tree***. Otherwise, it is called a ***non-active tree***. In Fig. 7 the output state is changed from high to low, so the P tree is the non-active tree and the N tree is the active tree. In general, for most CMOS logic structures, if one is an active tree, then the other will be a non-active tree; but for pseudo-nMOS logic this is not the case, the PMOS load is always turned-on, and therefore needs to be handled by another method. A tree staying in one of both states is dynamically dependent upon the input patterns, so we must decide the PN tree which is the *active tree* after each input event happens. From the experimental results, we have two rules for processing the internal nodes.

Rule 1:

Only the internal nodes in the *active tree* should be considered when estimating the delay time.

Rule 2:

The internal nodes in the *non-active tree* should be considered when estimating the slope, except that the RC time constant of the active tree is computed first.

For example, if the output state is changed from high to low in the circuit as shown in Fig. 4(a), there is at least one discharging path existing in the *active tree* (the N tree in this case) and no charging path in the *non-active tree* (the P tree). In this case, we must consider the effect of the nodes N_1,N_2,N_3, and N_4 when calculating the delay. On the contrary, the effect of the nodes N_5 and N_6 should be added when estimating the slope.

V. DELAY ESTIMATION

1. Overshoot

Owing to the electrical characteristics of a MOS transistor, there are many parasitic capacitors existing inside a CMOS gate, e.g., C_{gs}, C_{gd} and so on. The waveform of the drain of a MOSt depends not only the turn-on mechanism of MOSt but also on the path formed by C_{gd}. The overshoot of the output waveform, which can be treated as the excessive charge stored in the output node, is caused by the differential gate capacitor current. Observe that the amount of overshoot is determined by four factors as follows: (1) the slope S_i of the input signal, (2) the size of C_{gd}, (3) the load capacitance C_l of the output, and (4) the resistance R_p of the discharging path in the N tree (or charging path in the P tree). The structure and the processing method of both the N tree and P tree are identical, so hereafter only the N tree is discussed.

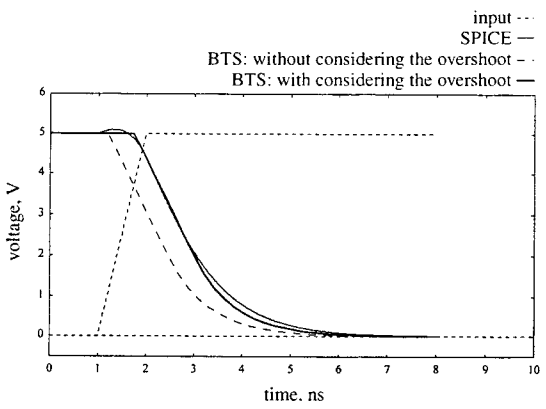


Fig. 8. Comparisons of simulated waveforms.

In Fig. 8 we can see that the overshoot is the major factor that produces the error between SPICE and BTS if we do not consider it into our simulator. In this case, BTS neglects the effect of C_{gd}, and uses the time that input voltage reaches V_T of a MOSt as the turn-on time. Therefore, the overshoot effect should be taken into account to obtain a more accurate transient waveform when we estimate the *delay*.

By analyzing some sample circuits using SPICE and varying the values of factors as mentioned above, we measure the data of delay time (not the width of overshoot) and then we can model the delay behaviors of CMOS gates by two equations.

(i) Dominant delay equation:

C_l is fixed, so this equation describes the relationship among switching delay, S_i, and R_p. Changing S_i is easy, but changing R_p is more difficult. Therefore, an alternative method is used. We increase the number of MOSt's in the N tree circuit in order to change R_p discontinuously, and then R_p is replaced with N_p. In other words, we use circuits such as inverter, two-input NAND gate, three-input NAND gate, and so on, as *the primitive cases*; but only one MOSt near the output accepts the input signal, and the others are kept in the turn-on state. The reason why the input signal must be placed near the output is so that the effect of internal charges can be avoided. The effect of internal charges we may meet in actual circuits are extracted as an independent problem (see next subsection).

The value of delay time is measured by three steps as follows. (1) Find two points whose voltages are 90% (4.5V) and 50% (2.5V) on the output waveform (for a falling signal). (2) Draw a straight line through these two points, and then produce a cross point, denoted by 'A' as shown in Figs. 5 and 6, when

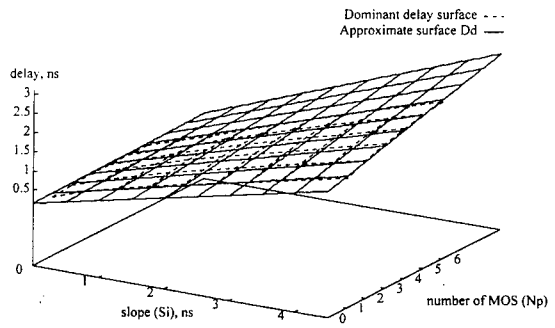


Fig. 9. The dominant delay surface and its approximate surface D_d.

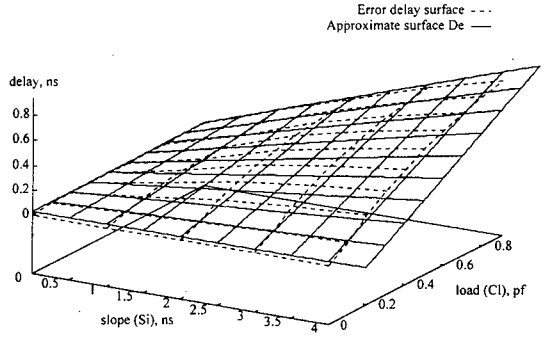


Fig. 10. The error delay surface (N_p=1) and its approximate surface D_e.

you meet the 5V horizontal line. (3) Measure the time distance between point A and the point that is the starting point (i.e. the point D in Figs. 5 and 6) of the overshoot. For each primitive case, changing the input slope will produce a set of discrete two dimension curves, called *NANDx-curves* (x is the number of input). By collecting all the sets of data, we can plot a three-dimensional surface as shown in Fig. 9. To simplify the calculation, we can use a hyperbolic surface (Eq. (12a), also shown in Fig. 9) to fit it. However, without obviously increasing the speed, the more accurate approximate surface (Eq. (12b)) is preferred to obtain a more accurate delay time.

$$\mathbf{D}_d=(0.0292N_p+0.369)(S_i+0.3)+0.12 \tag{12a}$$

$$\mathbf{D}_d=(-0.023N_p^2+0.19N_p+1.12)(-0.0047S_i^2 +0.38S_i+0.91). \tag{12b}$$

The deriving procedure is described as below:

Step 1: Use a straight line to fit a *NANDx-curve* in $\mathbf{D}_d$–S_i plane, called curve α.

Step 2: Use a straight line to fit the curve, called *SLOPEy-curve* (*y* is the value of the input slope), in the $\mathbf{D}_d$–N_p plane, and then normalize this curve, called curve β, which is used to modulate the curve α in the direction of the N_p-axis.

Step 3: $\mathbf{D}_d$=(curve α)(curve β)+offset.

This method is not the best for fitting the surface built by experimental data but it is very flexible and efficient. In general, the most events fall into the lower left corner area of the delay surface, and this procedure can let us specify the area to fit it better by choosing a *NANDx-curve* and a *SLOPEy-curve*. The offset is used for shifting the $\mathbf{D}_d$-surface to further approach the specified area because the area near the $\mathbf{D}_d$-axis is also a low event density area.

(2) Error delay equation:

Np is fixed, so this equation describes the relationship among error delay (an offset value with respect to $\mathbf{D}_d$, i.e., the $\mathbf{D}_d$ component is not involved), S_i, and C_l. This equation is used for compensating the value of delay time calculated by the dominant delay equation, which does not consider the effect of the changing factor C_l. If N_p is adjusted, we obtain a set of surfaces. It means that we can obtain a discrete three-dimensional surface for each primitive case. The method for constructing this surface is the same as mentioned above. The difference is that S_i, and C_l are changed for each primitive case, and the data are the offset values with respect to the values measured for the dominant delay equation. For example, if S_i = 1ns and C_l = 0.2pf for a two-NAND gate, the value of delay time is measured and suppose it is 0.6ns. However, the dominant delay is 0.5ns (also an assumed value) under the circumstance that Si = 1ns. Remember that C_l is always fixed when deriving the $\mathbf{D}_d$. Here suppose it is 0.1pf. Therefore, the error delay is 0.1ns, and this value is caused by C_l changed from 0.1pf to 0.2pf. Similarly, we can also use a set of hyperbolic surfaces

$$\mathbf{D}_e=f(N_p)(0.293S_iC_l+0.023) \tag{13}$$

to fit them, where $f(N_p)$ represents the coefficients that are the function of N_p. The surfaces when N_p=1 are shown in Fig. 10, which include the surface derived from the experimental data and its approximate surface.

In BTS the estimation of N_p is not easy in the simulation of real circuits because there may be several discharging paths. Therefore, an alternative method is adopted: N_p is calculated by the total equivalent resistance R_{peq} of all discharging paths divided by the turn-on-resistance R_{on} of single MOSt , i.e. $N_p \cong R_{peq}/R_{on}$. For example, in Fig. 4(a) let

$V_b=V_c=V_i=V_j=0V$ and $V_a=V_d=V_e=V_f=V_g=V_h=V_k=V_l=V_m=V_n=5V$, then $R_{peq}=R_h+(R_k||R_l)+R_m+R_n$. Although there are five turn-on NMOSt's, N_p is 3.5 in this case if all NMOSt's are identical. In the most cases, N_p will not be exactly an integer because of two reasons : (1) parallel-connection almost always exists in the gate circuits, and (2) there may be more than one MOSt situated in the transient (*off*-to-*on*) state in the CMOS gates. An active tree has at least one transient MOSt, and the transient MOSt that is most near the output node, denoted by MOSto, must be treated as a turn-on MOSt because one transient MOSt is always contained in the primitive cases (i.e., *i*-input NAND gate, *i*=integer number). It means that the transient MOSt, except MOSto, will be replaced with the effect resistance R_t. Because R_t is greater than R_{on}, N_p will be greater than the actual number of MOSt's in the discharging paths connected by series when more than one MOSt is in the transient state. It is reasonable because the larger resistance of R_t can be seen as the resistance composed of more than one MOSt that is in the turn-on state. Finally, we conclude that the effective resistances of transient MOSt's on the discharging path can be obtained by looking up the R_t-table but the one that is most near the output should be treated as the turn-on MOSt and replaced by R_{on}. Calculating the values of R_{peq} for each node in the PN tree is not extra work because it has already been built into BTS for solving the internal charge problem[4] [5].

2. Internal nodes

The delay due to the internal charges can be calculated approximately as

$$\Delta t=\frac{Q}{\bar{I}}=\frac{Q}{V_s}2R=\frac{\int CdV}{V_s}2R \tag{14}$$

where $\bar{I}$ is the average current, V_S is the voltage swing, R is the effective resistance of the conducting path and Q is the charge stored in the internal nodes [11]. More than one internal node may be going to charge or discharge in the PN tree, and these nodes must be taken into account when calculating the switching delay. Thus, Eq. 14 is rewritten as

$$\mathbf{D}_i=\sum_i 2R_i\frac{Q_i}{V_s} \tag{15}$$

where R_i is the effective resistance of internal node N_i with respect to the ground, and Q_i is the charge stored in the internal node N_i. For example, consider the circuit in Fig. 4(a). If $V_a=V_c=V_d=V_e=V_g=V_h=V_j=V_k=V_l=V_n=5V$, $V_b=V_i=0V$, and $V_f=V_m=0V$ to $5V$, then the internal nodes N_1 and N_3 are considered because only both nodes have the charges to be discharged. The total delay time caused by the internal charges is $\Delta t=\Delta t_{N1}+\Delta t_{N3}$, where

$$\Delta t_{N1}=2R_{N1}\frac{Q_{N1}}{V_s}=2((R_k||R_l)+R_m+R_n)\frac{Q_{N1}}{V_s}$$

$$\Delta t_{N3}=2R_{N3}\frac{Q_{N3}}{V_s}=2(R_m+R_n)\frac{Q_{N3}}{V_s}.$$

After all, the total delay is summed up by the delay times caused by the effect of overshoot and the internal nodes (including the charge sharing effect [4][5].)

$$\mathbf{D}=\mathbf{D}_d+\mathbf{D}_e+\mathbf{D}_i. \tag{16}$$

VI. SLOPE ESTIMATION

If the output waveform can be treated as a simple RC waveform, then the parameter T in Eqs. (10) and (11) can be calculated by the equation: $T=(t_{50\%}-t_{10\%})/2=0.294RC$. In BTS we defined slope as the time spent by the signal voltage dropping one volt, i.e. in units of time/volt, and then $T=\mathbf{S}$ when $V_{dd}=5V$.

$$T=\mathbf{S}=0.294RC \tag{17}$$

1. Non-active tree effect

Because the falling signal of the output is affected not only by the N tree but also by some internal nodes connected to the output node in the P tree, the algorithm for computing the slope should consider both and then estimate the total effect. The effect of the internal nodes in the non-active tree can be demonstrated by the circuit, for example, Fig. 4(a). Initially, let $V_a=V_b=V_d=V_e=V_h=V_i=V_1=0V$ and $V_c=V_f=V_g=V_j=V_k=V_m=V_n=5V$, so there are two internal nodes charged to $5V$,which are N_5 and N_6 in the P Tree. Next, by changing V_a, V_b and V_h from 0V to $5V$, a new discharging path is formed and the output will be discharged to *Gnd*. The discharging path is M_h-M_k-M_m-M_n-*Gnd*, where M_i is the MOSt whose input is named *i* and '-' means that the discharging current will flow from the left side to right side. If $V_e=0V$ before V_a, V_b and V_h are changed, N_5 and N_6 will be discharged to output through the path M_d-M_e-*output* and M_e-*output*, respectively. Observing the waveforms of nodes N_5, N_6 and the output node resulting from SPICE, we can find that the slope of N_5, N_6 and the output waveform are almost identical (see Fig. 11). Therefore, we estimate the waveform slope of

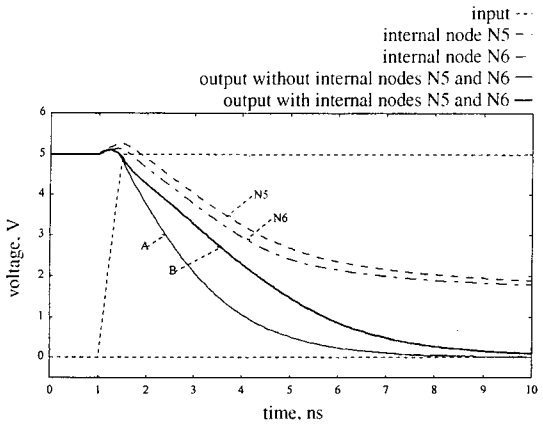


Fig. 11. Comparisons of output waveforms of the circuit in Fig. 4(a).

N_6 instead of the output node as the output slope. For comparison, we also consider the output waveform without it being affected by the internal nodes in the *non-active* tree. If we let V_e=5V before V_a, V_b and V_h are changed, then the internal nodes N_5 and N_6 will not participate in the discharging process. We also show this waveform in Fig. 11 (curve A).

2. Bottle neck effect

When estimating the slope, one of the most important factors, called bottle-neck effect, cannot be neglected. In fact, a bottle-neck always exists in the charging/discharging path, and dominates the charging/discharging rate. It easy to estimate the case where the circuit is in a pure series connection and only one MOSt is in the transient state. In this case, the MOSt is nearest the Gnd (ground) node, in general, there will be a bottle-neck if all MOSt are identical. However, the problem will become difficult to handle when multiple active input signals occur in a cluster, especially when the circuit is a complex gate. 'Multiple active input signals' means that more than two events make their correspondent MOSt's turn on simultaneously. If we replace all the transient MOSt's with *Rt* unconditionally, then a larger error will appear in the slope estimation. The error is caused by overestimating the influence of the transient MOSt's not in the bottle neck. The bottle neck effect and the multiple active input signals problem have been solved, see Ref. [6].

3. The algorithms

To simplify the algorithms, we modified the P tree by flipping it horizontally. Consequently, the P tree and N tree have the same direction when traversing them, namely, if we traverse them from left to right, it also implies that we traverse the P tree and N tree from the output node to the *source* node (V_{dd} or *Gnd*); refer to Fig. 4(b) to check this property of the PN tree. Based on the modified PN tree, we thus can develop a general form of the algorithm described below. Four steps are used in BTS for estimating the slope:

Step1: Traverse the *active tree* with a forward-post-order manner to calculate the total equivalent RC time constant $\tau 1$ of all discharging/charging paths (see algorithm ActiveTreeSlope( )). Note that the 'forward'-post-order means that the tree has been traversed from left to right.

Step 2: Traverse the *non-active tree* with a forward-post-order manner to calculate the equivalent RC time constant of each internal node connected to the output, and choose the node N_{max} with the maximum RC value $\tau 2$ (see algorithm NonActiveTreeSlope( )).

Step 3: Use the equation as follows to obtain the equivalent RC time constant τ.

$$\tau=\tau 1+\tau 2+(the\ R_{eq}\ of\ Active\text{-}Tree) \cdot (the\ C_{eq}\ of\ N_{max}). \quad (18)$$

The derivation of Eq. 18 is explained at the end of this section.

Step 4: Use the Eq. 17 to obtain the final result, that is, **S**=0.294τ. This **S**, exactly the T, can be substituted into Eqs. 10 and 11 to plot the transient waveform.

The equivalent RC time constant of the *active tree* can be computed by the equations as follows and by the recursive way while traversing the whole RC tree [2].

(1) leaf node:

$$R_{eq}=R_t \text{ or } R_{on}$$

$$C_{eq}=C$$

$$\tau_{eq}=R_{eq}C_{eq}. \quad (19)$$

(2) for series connection:

$$R_{eq}=R_{eq1}+R_{eq2}$$

$$C_{eq}=C_{eq1}+C_{eq2}$$

$$\tau_{eq1}=R_{eq1}C_{eq1} \quad (20)$$

$$\tau_{eq2}=R_{eq2}C_{eq2}$$

$$\tau_{eq}=\tau_{eq1}+\tau_{eq2}+R_{eq1}C_{eq2}.$$

Note: R_{eq2} is closer to the output node than R_{eq1}.

(3) for parallel connection:

$$R_{eq}=R_{eq1}||R_{eq2}$$

$$C_{eq}=C_{eq1}+C_{eq2}$$

$$\tau_{eq}=R_{eq}(\tau_{eq1}/R_{eq1}+\tau_{eq2}/R_{eq2}). \quad (21)$$

The same equations as mentioned above can be applied to step2 for the *non-active tree*, except that we process each node connected to the output individually (not the whole *non-active tree*) and that R_{eq2} is closer to the output node than R_{eq2}. The algorithms are listed as below:

```
EstimateSlope()
{
 for each block{
      if(output changed from LOW to HIGH){
            ActiveTreeSlope(*PtreePtr, &R1, &C1,
            &T1);
            NonActiveTreeSlope(*NtreePtr, &R2,
            &C2, &T2);
      }
      else{
            ActiveTreeSlope(*NtreePtr, &R1, &C1,
            &T1);
            NonActiveTreeSlope(*PtreePtr, &R2,
            &C2, &T2);
      }
      Search the vertex that has the max RC value in
      the NonActiveTree LinkedList, which is pro-
      duced by the NonActiveTreeSlope(), and copy
      the results to R2, C2 and T2.
      slope=((T1+R1*Cload)+(T2+C2*R1)) *0.294;
 }
}
ActiveTreeSlope(VERTEX *vertex, float *R, float
*C, float *T)
{
 if vertex is a leaf vertex{
      *R=Ron or Rt of vertex;
      *C=capacitance of vertex;
      *T=(*R)*(*C);
      return;
 }
 ActiveTreeSlope(left subtree of vertex, &leftR,
 &leftC, &leftT);
 ActiveTreeSlope(right subtree of vertex, &rightR,
 &rightC, &rightT);
 if vertex is an AND vertex{
      if(left subtree is nonactive or right subtree is
      nonactive){
            *R=INFINITE; *C=leftC;
            *T=0.0;
      }
      else{
            *R=leftR+rightR; *C=leftC+rightC;
            *T=rightT+leftT+rightR*leftC;
      }
 }
 else{  /* vertex is an OR vertex/
      case1: left subtree and right subtree are
      nonactive
            *R=INFINITE;      *C=leftC+rightC;
            *T=0.0;
            return;
         case2: left subtree is nonactive and right
         subtree is active
            *R=rightR;   *C=leftC+rightC;
            *T=rightT+rightR*leftC;
            return;
         case3: left subtree is active and right subtree
         is nonactive
            *R=leftR;     *C=leftC+rightC;
            *T=leftT+leftR*rightC;
            return;
         case4: left subtree and right subtree are ac-
         tive
            *R=(leftR*rightR)/leftR+rightR);
            *C=leftT/leftR+rightT/rightR;
            *T=(*R)*(*C);
            return;
 }
}
NonActiveTreeSlope(VERTEX *vertex, float *R,
float *C, float *T)
{
 if vertex is a leaf vertex{
      *R=Ron or Rt of vertex;
      *C=capacitance of vertex;
      *T=(*R)*(*C);
      return;
 }
 if vertex is an AND vertex{
            NonActiveTreeSlope(left subtree of ver-
            tex, &leftR, &leftC, &leftT);
            if(left subtree is nonactive){
                  *R=INFINITE; *C=0.0;
                  *T=0.0;
                  return;
            }
            NonActiveTreeSlope(right subtree of
            vertex, &rightR, &rightC, &rightT);
            if(right subtree is nonactive){
               allocate an element to record leftR,
               leftC and leftT , and then append it
               to NonActiveTreeLinkedList.
            *R=INFINITE; *C=rightC;
            *T=0.0;
```

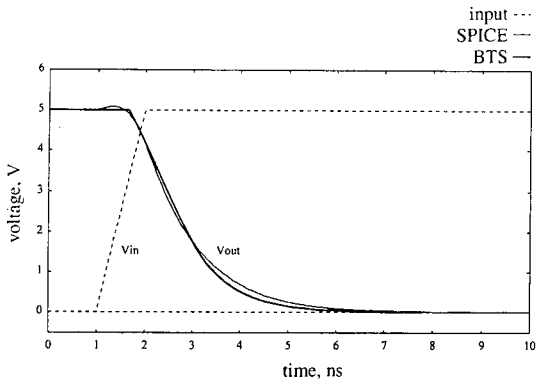


Fig. 12. Simulated waveforms of the complex gate in Fig. 4(a). Dot line: input.

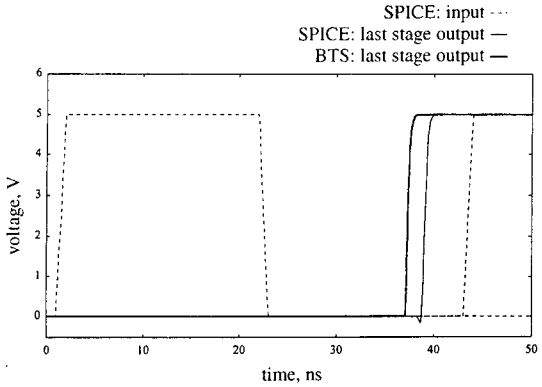


Fig. 13. Simulated waveforms of an inverter chain.

```
    else{
        *R=leftR+rightR; *C=leftC+rightC;
        *T=rightT+leftT+leftR*rightC;
    }
}
else{   /* vertex is an OR vertex/
        NonActiveTreeSlope(left subtree of vertex,
        &leftR, &leftC, &leftT);
        NonActiveTreeSlope(right subtree of ver-
        tex, &rightR, &rightC, &rightT);
        case 1: ...
        case 2: ...
        case 3: ...
        case 4: ...
        /* These four cases are the same as
        ActiveTreeSlope(); */
}
}
```

Equation (18) is derived by the same idea as Eq. (20): the whole *active tree* acts as R_{eq1}, and the path connecting N_{max} and the output node acts as R_{eq2}. In addition, both are connected in a series. Hence, we can use Eq. (20) to derive Eq. 18.

4. An example

To be more specific, an example smilar to one mentioned above (only the case that N_5 and N_6 participate in the discharging process, also see Fig. 4(a)) is used for demonstrating this method. In step1, the discharging path is M_h-M_k-M_m-M_n-*Gnd*, and this is a series connection. We traverse the N tree by the forward-post-order method, and use Eq. (20) to obtain the result $\tau 1=(R_hC_{load}+(R_kC_{N1}+R_mC_{N3}+R_mC_{N1})+(R_k+R_m)C_{load})+ R_nC_{N4}+R_n(C_{load}+C_{N1}+C_{N3})$. In step2, the discharging path is M_d-M_e-*output*, and this is also a series connection. We also traverse the P tree by the forward-post-order way, and obtain the result $\tau 2=R_eC_{N6}+R_dC_{N5}+R_eC_{N5}$. Finally, the total slope $\tau=\tau 1+\tau 2+(R_n+R_k+R_m+R_n)(C_{N6}+C_{N5})$.

The method in step 2 is an approximate method. However, it can meet our requirements in the most cases.

VII. SIMULATION RESULTS

This method has been tested extensively for basic modules such as counters, decoders, adders, and ALU's. A one-cluster circuit, also using the circuit as shown in Fig. 4(a), is simulated by using SPICE and our timing simulator BTS. The results are compared as shown in Fig. 12. The bold solid line is the result obtained from our simulator. There is only a small error presented in this circuit because it is an one stage circuit. The simulated waveforms resulting from BTS are plotted by using Eqs. (10) and (11). Of course, before plotting the waveforms, we should estimate the delay time and slope by using the methods stated in section V and VI. Actually, the waveforms are processed by the waveform-processor, which is a post-process; it means that only the values of delay and slope are calculated during the simulation. Fig. 13 shows the waveform comparisons of an inverter chain with SPICE results. In this circuit, there are 100 stages passed from input to output, and any error in delay estimation will be accumulated to the next stage. However, only a small total error (4.5%) is presented because an inverter is the simplest primitive case, and it does not contain any internal node. The simulated waveforms of an adder SN7483 and a four-bit ALU 74381 are shown in Fig. 14 and Fig. 15, respectively. A slightly larger amount of error is produced, chiefly due to internal charges, input patterns (different from the primitive cases) and accumulated error, but the largest error is no more than 10% in both cases.

The CPU time comparisons are summarized in Table 1. The results show that the speed of BTS is two or three-orders faster than that of SPICE if the

Table 1 : Comparisons between BTS and Spice

Circuit	MOSt no.	CPU time on PC (DX4-100), secs BTS	Pspice	Speed ratio	Primary input event no.	Max. error
five-input NAND gate	10	0.50	29.33	0.017	40	< 1%
complex gate (Fig. 4(a))	14	0.11	2.53	0.043	2	< 1%
inverter chain (100 stages)	200	0.28	241.18	0.0012	1	4.5%
7483	258	0.99	774.64	0.0013	13	6.9%
74381	584	1.10	1670.98	0.00066	14	4.4%

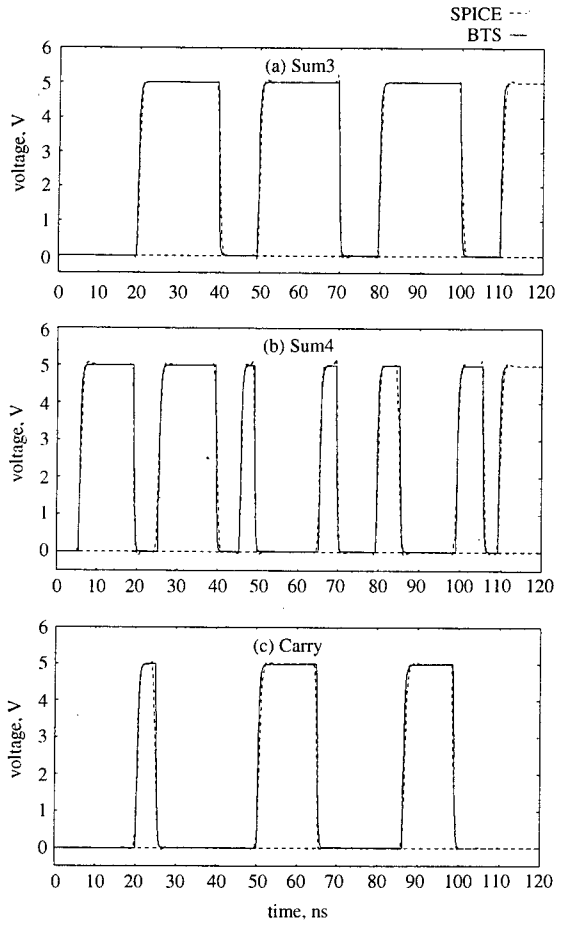


Fig. 14. Simulated waveforms of a 4-bit binary full adder with fast carry SN7483A. (a) sum output bit3, (b) sum output bit4, and (c) carry output.

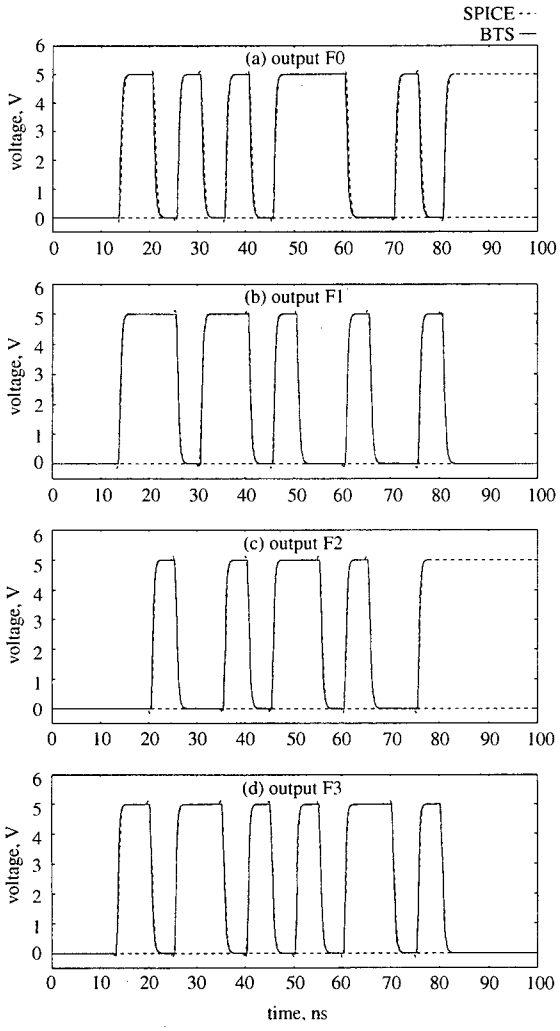


Fig. 15. Simulated waveforms of a 4-bit ALU SN74S381.

circuit is a large circuit. The speed ratios (simulation time of BTS divided by that of SPICE) are also shown in Table 1. It is a little bit slower than the previous version of BTS; due to the fact that the delay model in the new version of BTS is more complicated than it was in the previous version of BTS.

VIII. CONCLUSION

An accurate waveform approximation technique is proposed, and achieved by a new approach of delay estimation and modified slope estimation. This

method improves the previous version of BTS that converted the overshoot effect to the turn-on-time of MOSt (the value of V_T was shifted to 3.1V), and can offer better adaptability for a wide range of circuit and input specifications. For each different fabrication process, the equations for delay estimation are derived only once. The deriving procedure is simple and quick, not tedious work because it can be achieved by only a few samples. Of course, this procedure can also be done by a computer program.

The multiple active input signals will affect the delay and slope behavior of CMOS circuits. The latter problem, where slope behavior is affected by multiple active input signals, has been conquered [6] and the former problem, where delay behavior is affected by multiple active input signals, is the future work. In addition, the relationship among the delay, distribution and amount of the internal charges should be researched further.

IV. ACKNOWLEDGEMENT

This research was supported in part by the National Science Council of the Republic of China, under Contract No. NSC85-2215-E-002-020.

NOMENCLATURE

C	capacitor
C_k	parasitic capacitance at node k
C_{load}	load capacitance
$\mathbf{D}$	delay
$\mathbf{D}_d$	dominant delay
$\mathbf{D}_e$	error delay
Gnd	ground
M_i	the MOSt whose input is named i.
Q	charge
R	effective resistance
R_{eff}	effective resistance
R_{on}	resistance of MOSFET at high input
R_t	resistance of MOSFET during gate voltage from low to high
$\boldsymbol{S}$	slope
S_i	slope of the input waveform
T or t	time
Vdd	supply voltage
V_T	threshold voltage

Greek symbols

$\Delta\tau$	charging/discharging time
τ	time constant

References

1. Bryant, R. E., "A Switch Level Model and Simulator for MOS Digital Systems," *IEEE Transactions on Computers*, Vol. C-33, pp. 160-177 (1894).
2. Caisso J. -P., E. Cerny, and N. C. Rumin, "A Recursive Technique for Computing Delays in Series-Parallel MOS Transistor Circuits," *IEEE Transactions on Computer-Aided Design*, Vol. 10, No.5, pp.589-595 (1991).
3. Chang, F.C., C.F. Chen, and P. Subramaniam, "An Accurate and Efficient Gate Level Delay Calculator for MOS Circuits," Proceedings of 25th ACM/IEEE conference on Design automation, Anaheim, CA, USA, pp. 282-287 (1988).
4. Chang, M., S,-J Yih and W. S. Feng, "Algorithm Based on Modified Threaded Binary Tree for Estimating Delay Affected by Internal Charges in CMOS Gates", *Electronics Letters*, Vol. 32, No. 20, 26th September 1996, pp. 1877-1879 (1996).
5. Chang, M., S,-J Yih and W. S. Feng, "Recursive Algorithm for Calculating Effective Resistances in RC Tree", *Electronics Letters*, Vol. 33, No. 2, 16th January 1997, pp. 131-133 (1997).
6. Chang, M., S, -J Yih and W.S. Feng, "The Estimation of Bottle Neck Effect in Waveform-Based Switch-Level Timing Simulation", The 8th VLSI Design/CAD Symposium, Nantou, Taiwan, Aug. 21-23, pp. 233-236 (1997).
7. Elmore, W. C. "The Transient Response of Damped Linear Networks with Particular Regard to Wide-Band Amplifiers," Journal of Applied Physics, Vol. 19, No. 1, pp. 55-63 (1948).
8. Lin T. M., and C. A. Mead, "Signal Delay in General RC Networks," *IEEE Transactions on Computer-Aided Design*, Vol. CAD-3, No. 4, pp. 331-349 (1984).
9. Rubinstein, J., P. Penfield, and M. A. Horowitz, "Signal Delay in RC Tree Networks," *IEEE Transactions on Computer-Aided Design*, Vol. CAD-2, NO. 3, pp. 202-211 (1983).
10. Terman, C. J. "RSIM - A Logic-Level Timing Simulator,"Proceedings of the IEEE International Conference on Computer Design, New York, pp. 437-440, Nov. (1983).
11. Wang, J. H., M. Chang, and W. S. Feng, "The Effects of Internal Charges to Waveform Calculation," ASIA-PACIFIC Conference on Circuits and Systems, Australia, pp. 111-115 (1992).

Discussions of this paper may appear in the discussion section of a future issue. All discussions should be submitted to the Editor-in-Chief.

Manuscript Received: May 14, 1997
Revision Received: Sep. 26, 1997
and Accepted: Mar. 07, 1998

開關階層時序模擬器 BTS 之波形近似技術

張茂林　王志恆　易序忠　馮武雄

國立臺灣大學電機工程學研究所

摘　要

開關階層時序模擬器有快速與適用於VLSI電路的優點，但是卻無法提供精確的波形資訊。本文提出一準確且具效率的開關階層時序模擬器。高準確性係源自一新的波形近似技術，此技術包含延遲評估與斜率評估。有效率的延遲與斜率之計算則是以開關階層模擬取代電晶體階層模擬而達成。延遲評估的新方法將在文中描述：此法以二個方程式來模型化RC樹的延遲行爲，而此二式分別稱之爲主要延遲方程式與誤差延遲方程式。二者皆以曲面擬合（surface fitting）去近似量自CMOS閘之實際延遲行爲所產生之曲面。另外，斜率評估的方法也作了進一步的修改。斜率與電路之等效RC時間常數有著密切的關係。而等效RC時間常數可以經由對RC樹作遞迴式的拜訪而獲得。模擬結果與SPICE比較，有令人滿意的結果。

關鍵詞：時序模擬，波形近似技術，RC樹。

EXPLORING THE DESIGN SPACE OF CACHE MEMORIES, BUS WIDTH, AND BURST TRANSFER MEMORY SYSTEMS

Chung-Ho Chen
Department of Electronic Engineering
National Yunlin University of Science and Technology
Yunlin, Taiwan 640, R.O.C.

Key Words: burst transfer memory, cache system, data path, die area, performance tradeoffs.

ABSTRACT

Caches, data path, and burst transfer memory are the major hardware techniques used to reduce the latency between the processor and the main memory. We explore the design space among the hit ratio (hence a cache size, or an improved cache structure), data path width, and the transfer memory design through a performance tradeoff methodology. For the tradeoffs among these three factors, our evaluation shows that if a D-byte data path system and a 2D-byte data path system have the same performance, then the hit ratio difference that trades the performance of a D-byte wide data path is between 0 (low bound) and *1-HR* (high bound) where *HR* is the hit ratio associated with the D-byte system. For current main memory systems, doubling the data path trades about half of the high bound of the hit ratio traded in a transfer-time dominated system. Doubling the data bus is more advantageous when the processor is designed with the use of a high-speed non-constant-time-dominated L2 cache. Doubling the bus width trades a large percentage of the hit ratio when a large amount of non-cacheable 2D-byte memory traffic exists.

I. INTRODUCTION

IC technology has accelerated processor speed in the range of hundreds of MHz. To cope with such a high-speed processor, a second-level cache memory (L2), a larger processor data bus, or a faster memory is used to achieve a higher performance [1, 6, 9, 10]. In a cache-based system, both the cache system and wider data path reduce the average memory delay time and thus improve performance. One system using a larger cache may claim better performance over a system using a wider data path and vice versa [6]. The same argument also applies to performance using either a larger cache or a faster memory. Using an on-chip second-level cache or a larger primary cache increases the die cost while a wider data bus increases the processor pin count and the packaging cost. A small CPU packaging size is quite desirable in hand-held machines. The performance of a faster memory and a higher bus transfer rate is usually limited by the reliability of transfers among devices [1]. When coming to the decision to increase the data path, the transfer speed, or the die area of the on-chip L1 cache to improve performance, the priority of which should go first is unclear and often quite contentious. However, it is well known that these hardware techniques all contribute to the performance of a cache-based system [6].

A fair way to determine which architecture alternative should be given higher priority is to examine the costs or design complexity based on a performance equivalent point for which one alternative is used while the other is not. The configuration that has a lower implementation cost or less design complexity but with the same performance is then the best choice. If we examine how cache memories, data path

width, and memory design affect performance, we obtain the following observations. Improving the cache hit ratio reduces the mean memory delay time [4]. Increasing the width of the data bus (processor data path or board level bus) or using a higher speed memory also reduces the mean memory delay time. Therefore, we can establish a performance equivalent point based on the mean-memory delay time [2]. Once a certain cache size, data path width, and memory design are decided with the desired performance point, the system configuration with a lower cost or less implementation complexity can be easily determined.

In this paper, we assess the design tradeoffs among the cache hit ratio (hence a cache size, or an improved cache structure), data path width, and burst transfer memory design with a performance model based on the mean memory delay time. The memory cycles of reading a cache line in a burst transfer memory system consist of a constant time and a per-bus transfer time. We show how various transfer memory designs affect the amount of hit ratio traded with a wider data path. In particular, if a D-byte system and a 2D-byte system have the same performance, then the hit ratio difference that trades the performance of a D-byte wide data path is between 0 (low bound) and 1-*HR* (high bound) where *HR* is the hit ratio associated with the D-byte system. The exact amount of hit ratio traded is determined by the constant time and the per-bus transfer time of the memory system used. For current main memory systems, doubling the data path trades about half of the high bound of the hit ratio traded in a transfer-time dominated system. Doubling the bus width trades a large percentage of the hit ratio when a large amount of non-cacheable 2D-byte memory traffic exists.

Applications like hand-held machines usually require a small CPU packaging size. For these systems, a data path width of a 4-byte system with a cache of less than 32KB is quite attractive because it achieves the performance level of an 8-byte system with a cache of up to 8KB for the SPEC92 benchmarks. Doubling the bus width trades a hit ratio that is close to the high bound if the processor is connected to a transfer time dominated memory system. For this, doubling the data bus of a processor is more advantageous when the processor is designed with the use of a high speed non-constant-time-dominated L2 cache in mind. For the SPEC92 benchmarks, we observe that doubling the bus trades more die area when the existing cache size is large. In a constant time dominated system, the effectiveness of doubling the data path is limited if no non-cacheable 2D-byte memory references exist. In systems using a burst transfer memory design, an attempt to double the data path should be accompanied by a reduction of the constant time. Otherwise, a cache structure enhancement, such as a higher set associativity, can easily achieve the performance improvement of doubling the data path. However, if a large amount of non-cacheable 2D-byte memory traffic exists, doubling the bus width should be considered first.

The rest of this paper is organized as follows. Section 2 discusses the related work. The notation and assumptions of the hardware under study are given in Section 3. In Section 4, we develop the performance equivalence model, present the tradeoff results of hit ratio and data path, and examine the implications on the CPU die area with respect to the design of the memory system. Generalization of the model to include non-cacheable memory references is presented in Section 5. We discuss issues regarding the validation of the model in Section 6. Our conclusions are given in Section 7.

II. RELATED STUDY

Design tradeoffs for two-level caches have been evaluated in [1, 6]. Jouppi combined Mulder's area model and Wada's access time model with miss ratios to determine better on-chip memory configurations [7, 12]. Data path width and bus transfer speed are not considered as a tradeoff alternative in the study. The bandwidth of data memory hierarchy for superscalar processors has been examined in [11]. Sohi suggested that L1 cache design incorporate cache properties such as non-blocking and multi-ported for superscalar machines with a high issue rate. Smith and Przybylski used the cache miss ratio obtained from trace-driven simulations to study the factors for choosing a cache line size [8, 10]. The criterion in selecting the best line size is to find the line size which minimizes the mean memory delay time per memory reference or the mean read time [10]. Bugge examined two-level cache designs by calculating the cache miss cost, which is basically the miss ratio multiplied by the miss cycles in refilling a line as that used by Smith [1]. From these previous studies, we found that the mean memory delay time metric has been used extensively in addressing many aspects of the design of a cache-based memory system [4].

In our earlier work, we used the mean memory delay time metric to evaluate the performance of various hardware techniques [2]. The results showed that the pipelined memory system is most advantageous for increased performance, while doubling the bus width is the second choice and the use of read-by-passing write buffers comes next. In the earlier research we used the non-pipelined memory model as the baseline system to compare the performance of various hardware techniques. The results presented in this paper are an extended version of Section 6.2

in [2]. We address new issues uncovered in that research including the implications of using L2 caches and transfer memories on the tradeoff of the cache hit ratio, results of using SPEC92 benchmarks, and the generalization of the tradeoff model for non-cacheable memory references.

III. SYSTEM MODEL

We consider a RISC-type processor that has split caches. The data cache uses the write-back and write-allocate policy and brings in a whole line from the memory for a cache miss. The memory system is assumed to use the same memory cycle time for read/write misses. Table 1 illustrates the architecture and application parameters for a typical cache-based system where the number of bytes reads, R, can be related to the hit ratio of a cache (i.e., its size). Parameter α is used to specify the amount of copy-back traffic due to replacement of dirty lines. For a cache-based system, we distinguish cache-stalling features that are related to various cache implementations, and processor bus interface control that is related to the memory system design. A cache miss can be satisfied in different ways that are determined both by the cache implementation as well as the memory system design.

For instance, a designer can simplify the memory system design by using a non-pipelined memory system. Alternatively, the designer can design higher performance memory systems such as pipelined or burst transfer memories in which the first read takes more cycles and the subsequent words come at a faster rate. However, the cache and processor implementation determines when the processor may use the requested data. As an example, the first read may fetch the first word in a cache line or the first read may fetch the requested word of the processor. The processor may be stalled until the entire line has been filled or it may start execution as soon as its requested word has come into the cache. A non-blocking cache is a more aggressive implementation that allows the CPU to continue during a miss. These alternatives are cache stalling features. That is, a burst transfer or pipelined memory is a mechanism which moves a cache line into the cache faster while the stalling features determine when the CPU may use the available cache data. Thus, irrespective of the cache stalling features, a general form of memory cycles of reading a cache line in a burst transfer memory can be represented by $C+\beta\,(L/D)$ where C is a constant time and β is the number of clocks of per-bus transfer. Constant time C is the clock cycles for the presentation of address to the memory system and memory access latency. It may include cycle times for address decoding and bus arbitration.

Table 1. Architecture and application parameters.

L	Cache line size in bytes.
D	Processor external data bus or memory data path width in bytes.
$C+\beta(L/D)$	Memory cycle time per cache line normalized with the CPU clock cycle.
E	The number of cycles executed by a processor for an application.
R	The number of data bytes read in full bus width by a processor upon read or write misses for E cycles executed.
α	Cache line copied-back ratio for E cycles executed ($0\le\alpha\le1$).
E_i	The total execution cycles for all internal (non-memory reference) instruction
E_m	The number of load/store instructions.
h_t	Cache hit cycle time normalized with the CPU clock cycle.

IV. TRADE-OFFS OF CACHES AND TRANSFER MEMORY DESIGN

In this section, we first quantify the CPU execution time for the out-of-order execution processor model and develop the performance equivalent point where various architectural features can be used to achieve that. Then we present the trade-offs among these hardware techniques and address the implications on practical issues.

1. Components of execution time for out-of-order processors

Performance is best quantified by the program execution time. For current out-of-order execution processors, quantifying the execution time is a profound and difficult task. We approach this problem by classifying the execution time into different components. The first portion of the CPU execution time consists of the cycles when the processor executes the non-memory reference instructions or memory reference instructions that are on-chip cache hits. This portion of the cycle time is due to the processor chip's internal activities. The second part of the CPU execution time is due to executing instructions that have bus requests. This portion comes from cache misses for reads and writes and replacement of dirty cache blocks. Total execution cycles can be represented by E as follows:

$$E = E_i \cup (E_m - \frac{R}{L})h_t \cup (\frac{R(1+\alpha)}{L})(C + \beta\frac{L}{D}) \qquad (1)$$

where E is the union of three major cycle counts, that is, from the total execution cycles for non-memory instructions, the cache hit cycles, and the external memory reference cycles. These three cycle counts may overlap each other depending on the hardware used, especially for current dynamic execution superscalar processors. In Eq. (1), $(E_m-R/L)h_t$ represents the cycles of load/store instructions that hit in the data cache since R/L represents the number of load/store instructions that cause cache misses. We do not represent any cycles due to instruction fetching because they are assumed to be overlapped with E. If instruction fetching contributes to the execution time, its effect is similar to the increase in R. For an in-order issue and in-order completion scalar processor, its execution time E_s can be simply represented as

$$E_s = E_i + (E_m - \frac{R}{L})h_t + (\frac{R(1+\alpha)}{L})(C + \beta\frac{L}{D}) \qquad (2)$$

In this case, we assume that the cache is full blocking and there are no write buffers. Therefore, the miss cycles and write-back cycles all contribute to the execution time. If we improve the system by using certain hardware mechanisms, such as the out-of-order execution model, a non-blocking cache, or read bypassing write buffers, or a combination of these, the net effect scales the execution time E_s with a speedup factor, sp. In this system, the improved execution time

$$E_D = (E_i + (E_m - \frac{R}{L})h_t + (\frac{R(1+\alpha)}{L})(C + \beta\frac{L}{D}))\,sp \qquad (3)$$

where $0<sp<1$. To focus on the tradeoffs among caching, bus width, and transfer memory design, we consider the cycle times represented by Eq. 3 of a speed up factor, sp. In this way, we isolate the effect of other architectural factors on the execution time except hit ratio, bus width, and transfer memories. We elaborate more on this issue in a latter section by considering the case that portion of the memory latency may be hidden from the CPU execution time.

2. Performance equivalent point

In a uniprocessor system, one can design the system to achieve the same performance (execution time) by using either the combination of a larger cache (better caching) with a smaller data path, or the combination of a smaller cache with a wider data path. For instance, if System A uses a D-byte data path while System B uses one 2D-byte wide, then System A must use a larger cache than System B for a performance equivalent point. Hence, there is a relationship between the difference of the hit ratio of the two systems and the data path width at the same performance equivalent point. Let the execution time for the system increasing its processor external bus as well as memory data bus to 2D bytes be represented as follows.

$$E_{2D} = (E_i + (E_m - \frac{R'}{L})h_t + (\frac{R'(1+\alpha')}{L})(C + \beta\frac{L}{2D}))\times sp\,. \qquad (4)$$

In the 2D-byte system, the cache line size, constant time, and per-bus transfer time are the same as those in the D-byte system. Also the number of instructions executed is the same because the same E_i and E_m are used. The differences are the bus width and the caching capability, i.e., the size of the caches or their structures. To establish a performance equivalent point between the D-byte system and the 2D-byte system, we let $E_D=E_{2D}$, and find the memory reference ratio r in Eq. (5).

$$r = \frac{R'}{R} = -\frac{(1+\alpha)(C+\beta\frac{L}{D}) - h_t}{(1+\alpha')(C+\beta\frac{L}{2D}) - h_t}\,. \qquad (5)$$

Let the number of load/store instructions that hit (miss) in the data cache be denoted by $\lambda_h(\lambda_m)$ and let $\lambda_h=s\lambda_m$. The miss ratio, MR_1, of the data cache for the D-byte system is given by

$$MR_1 = \frac{\lambda_m}{\lambda_m + \lambda_n} = \frac{1}{s+1}\,. \qquad (6)$$

We use the hit (miss) ratio in the D-byte system as a base, namely a given hit (miss) ratio for the D-byte system is used. To achieve the same performance, the 2D-byte system could afford a lower hit ratio than the hit ratio in the D-byte system. Equivalently, the 2D-byte system can use a smaller cache to have the same performance. Since the same application is considered, $\lambda_h+\lambda_m=\lambda'_h+\lambda'_m$. Only some load/store instructions that hit in the cache of the D-byte system become misses in the cache of the 2D-byte system due to a smaller cache size for the same performance point. By saying this, we assume that when the two systems are compared the cache line copy-back ratio is the same.

Let MR_2 (HR_2) be the miss (hit) ratio associated with the 2D-byte system, and $\lambda'_m=r\lambda_m$ where $\lambda'_m=R'/L$, then

$$MR_2 = \frac{\lambda'_m}{\lambda'_m + \lambda'_n} = \frac{r\lambda_m}{\lambda_h + \lambda_m} = \frac{r}{s+1}\,. \qquad (7)$$

Let HR_1 be the hit ratio associated with the D-byte system. For the two systems, the difference of the hit ratio equals the difference of the miss ratios. Then the hit ratio difference that trades the performance of

a D-byte width is

$$HR_1 - HR_2 = MR_2 - MR_1 = \frac{r-1}{s+1} \tag{8}$$

where $s = \frac{HR_1}{1-HR_1}$ (from $\lambda_h = s\lambda_m$.) and $r = \frac{R'}{R}$ ($\lambda'_m = r\lambda_m$). Eq. (8) is only valid for the physical system where $HR_2 \geq 0$. Also from $E_D = E_{2D}$, we have

$$\frac{R\frac{(1+\alpha)}{L}(C+\beta\frac{L}{D})+(E_m-\frac{R}{L})h_t}{\lambda_h+\lambda_m} = \frac{R'\frac{(1+\alpha')}{L}(C+\beta\frac{L}{2D})+(E_m-\frac{R'}{L})h_t}{\lambda'_h+\lambda'_m} \tag{9}$$

because $\lambda_h+\lambda_m = \lambda'_h+\lambda'_m$ for the same workload. Therefore, the performance tradeoff is based on the equivalence of mean memory delay time. This property allows us to evaluate the design tradeoffs among the hit ratio, bus width, and transfer memory design without complicating the problem with other architectural features. That is, if one feature has a higher performance in terms of mean memory delay time, this feature in general will also have a higher performance in terms of affecting the CPU execution time.

If we introduce instruction miss penalty into the numerator on each side of Eq. (9) and replace the denominator with $I+\lambda_h+\lambda_m$ and $I+\lambda'_h+\lambda'_m$ where I is the number of instruction references, the above trade-off model can be applied to a compositive hit ratio. Using this trade-off methodology, we can determine the bound of the hit ratio that trades the performance of doubling the data path width for a burst transfer memory system.

Theorem 1.

If a D-byte system and a 2D-byte system have the same performance, then the hit ratio difference that trades the performance of a D-byte wide data path is between *0* (low bound) and 1-HR_1 (high bound).

Proof:

For a transfer time dominated memory system (β is much greater than C), and given that $\alpha=\alpha'$, we apply L'Hospital's rule [5] in Eq. (5) and find $r = \frac{R'}{R}$ =2. Therefore, $HR_1-HR_2=1-HR_1$ from Eq. (8). On the other hand, for a constant time dominated memory system (C is much greater than β), and $r = \frac{R'}{R}=1$. This determines the low bound of the hit ratio, that is, $HR_1-HR_2=0$.

Corollary 1.

Using the hit ratio HR_2 of the 2D-byte system as a base, the amount of hit ratio difference that trades the performance of a D-byte bus width is between 0 and 0.5 (1–HR_2).

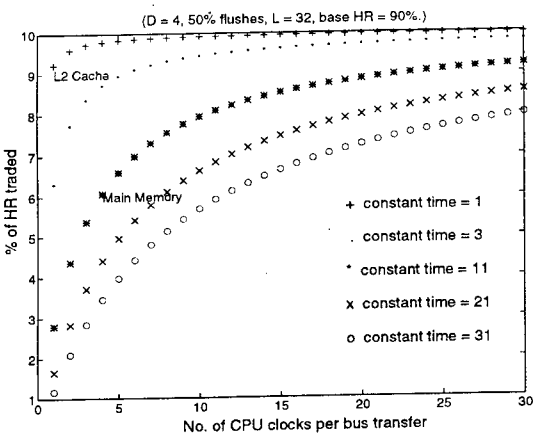


Fig. 1. Data Path and Hit Ratio Tradeoff Based on Burst Transfer Memories

Proof:

For a transfer time dominated memory system, using **Theorem 1**,

$$HR_1 - HR_2 = 1 - HR_1 = 1 - \frac{HR_2+1}{2} = 0.5(1-HR_2).$$

For a constant time dominated memory system, the low bound of the hit ratio difference is $HR_1-HR_2=0$ as before. Therefore, the amount of hit ratio that trades the performance of a D-byte data path is between 0 and 0.5 (1–HR_2).

The performance tradeoff between a 32-bit data path and a cache memory at a base hit ratio of 90% is illustrated in Fig. 1. When the processor external data bus and memory width are increased from 32 bits to 64 bits, the hit ratio in the 64-bit system can be smaller than the base hit ratio of the 32-bit system for both systems to have the same average memory delay time. The amount of the hit ratio traded is plotted on the y-axis in Fig. 1. The design limit is reached when the per-bus transfer time and the constant time have the value of one. We assume that the flush ratio is 0.5 although other values can also be used. In [9], Smith also used 50% in describing the copy back traffic. The cache hit time used is assumed to be 1 CPU clock cycle.

3. Connecting to a high speed L2 cache

When the processor is connected with a L2 cache, the constant time and transfer time are often relatively small. This design area is shown in the upper left corner of Fig. 1. Considering C=1, for

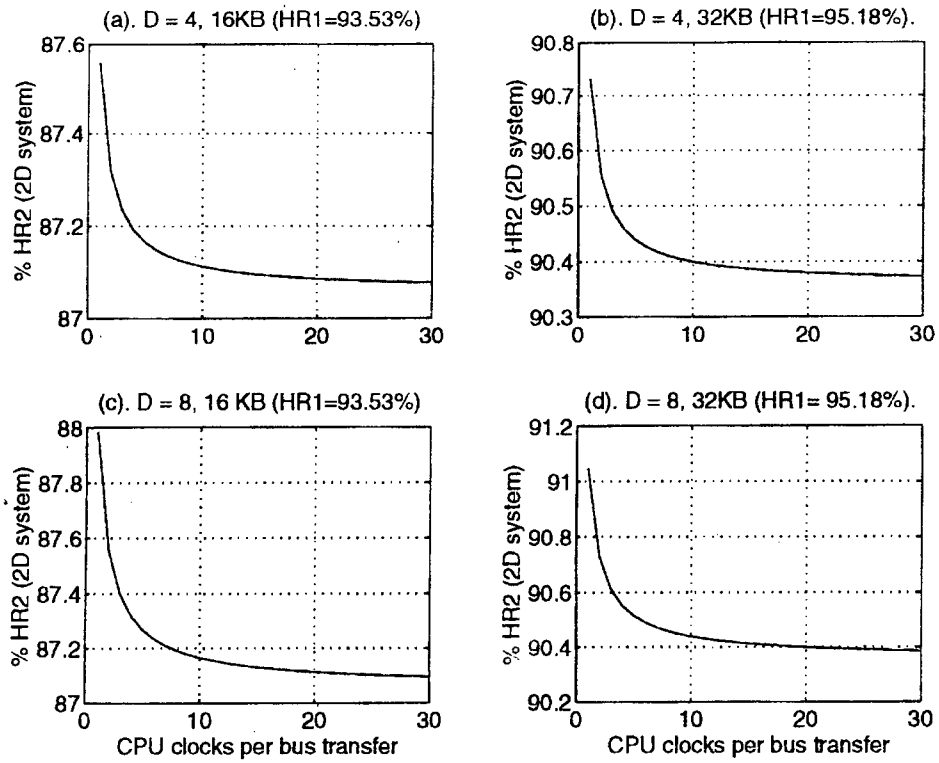


Fig. 2. The constant time is one CPU clock for connecting to a L2 cache. Performance equivalent systems between (a). (4-byte, 16KB) and (8-byte, HR2), (b). (4-byte, 32KB) and (8-byte, HR2), (c). (8-byte, 16KB) and (16-byte, HR2), and (d). (8-byte, 32KB) and (16-byte, HR2).

instance, a 32-bit bus system using a cache with a hit ratio of 90% has the same performance as a 64-bit bus system using a cache of the same line size with a hit ratio of about 80-81% (90–10 or 90–9, depending on the β used). In other words, increasing the hit ratio from 80% to 90% (10% increase), we can reduce the bus width (processor data bus and memory bus) from 64 bits to 32 bits while the same performance is preserved for a memory system of small constant time. The bound of the hit ratio traded for the performance of a 4-byte bus in this small constant-time memory system is 10%. This value can also be obtained by Theorem 1. That is, $HR_1-HR_2=1-HR_1=1-0.9=10\%$. The amount of hit ratio traded greatly depends on how the constant time and the transfer time are used. As an example, if the constant time = 3 and the transfer time = 1 as shown in Fig. 1, the amount of hit ratio traded for the bus width is close to half of the high bound. Because current high performance CPUs are usually designed in use with a L2 cache, the hit ratio that trades the performance of doubling the bus width will reside between half of the high bound and the high bound.

We map the cache hit ratio to cache size and show the cost and performance relationship between cache size and data path width. To explain this, we use the hit ratio data from the simulation results in [3]. The results are the average hit ratio of the SPEC92 benchmarks for a direct-mapped cache with 32-byte lines. These hit ratio data are reported in Table 2 again for the following explanation. Suppose that one considers improving the performance of a 4-byte data path and an 8KB on-chip data cache processor. One of the alternatives is to increase the cache size, for example, and keep the data bus pin count down for other purposes, such as Vcc and ground, and keep the CPU packaging size small. Small packaging size is highly attractive especially in hand-held computing machines. Increasing the cache to 16KB, for instance, is a viable choice. For the same performance level, however, the designer has another alternative to use, that is, using a 64-bit data path while a smaller cache may be considered when the die area is the constraint or the CPU packaging is not a concern. We consider the best possible design of using a L2 cache system in which C=1

Table 2. Hit Ratio (%) of Data Caches for the Average of the SPEC92 Benchmarks.

1KB	2KB	4KB	8KB	16KB	32KB	64KB	128KB
75.39	79.43	84.06	89.81	93.53	95.17	96.23	97.12

Table 3. Systems at performance equivalent point for the SPEC92 benchmarks. Both the constant time and the tranfer time are 1 CPU clock cycle. (The hit ratio of the 2D-byte system used is closer to the cache size indicated by the '+' sign.)

Base width = 4 bytes	Base width = 8 bytes
(4-byte, 4KB) = (8-byte, <1KB)	(8-byte, 4KB) = (16-byte, < 1KB)
(4-byte, 8KB) = (8-byte, 2^+-4KB)	(8-byte, 8KB) = (16-byte, 2-4KB)
(4-byte, 16KB) = (8-byte, 4-8KB)	(8-byte, 16KB) = (16-byte, 4-8^+KB)
(4-byte, 32KB) = (8-byte, 8^+-16KB)	(8-byte, 32KB) = (16-byte, 8-16KB)
(4-byte, 64KB) = (8-byte, 8-16^+KB)	(8-byte, 64KB) = (16-byte, 8-16^+KB)
(4-byte, 128KB) = (8-byte, 16-32KB)	(8-byte, 128KB) = (16-byte, 16-32KB)

and β=1. In such a system, the first bus access would require 2 CPU clock cycles while each of the subsequent bus accesses for the rest of the same cache line takes one CPU clock cycle. In Fig. 2(a), we show the hit ratio required for the 8-byte bus system to have the same performance as the (4-byte, 16KB) system. For β=1, the 8-byte system can use a cache of hit ratio 87.55% with the same line size to achieve the performance of the (4-byte, 16KB) system. Referring to Table 2, the cache size corresponding to 87.55% of the hit ratio is between 4KB and 8KB. That is, the (8-byte, 4-8KB) system delivers the same performance as the (4-byte, 16KB) system for the memory system used. It must be emphasized that the memory system used determines whether the (8-byte, 4-8KB) and the (4-byte, 16KB) system deliver the same performance.

Considering the range of a larger cache size, suppose that a (4-byte, 32KB) system is to be implemented. The 32KB cache has a hit ratio of 95.18% from Table 2. Using the same memory parameters, the alternative in the 8-byte counterpart with the same performance requires a cache with a hit ratio of about 90.72% as shown in Fig. 2(b). The mapping indicates about a size of close to 8KB but less than 16KB from the simulation data in Table 2. Thus, the (4-byte, 32KB) system and the (8-byte, 8^+-16KB) system deliver the same performance under the specified memory system.

Doubling the base bus width for comparison, we consider the case of an (8-byte, 16KB) and a (16-byte, HR2) system. In Fig. 2(c), for the same memory parameters, the hit ratio for the equivalent performance is about 88% for the 16-byte system. This would correspond to a size of close to 8KB but greater than 4KB. While considering the case of (8-byte, 32KB) and (16-byte, HR2) as shown in Fig. 2(d), we find that the hit ratio for the equivalent performance is 91.05% for the 16-byte system. This maps to a cache size of between 8KB and 16KB. Similarly, we can obtain results for other cache sizes. These results and the above are summarized in Table 3. Current process technology has been able to put more than 60% of the die area to the on-chip caches such as the R10000 processor [13]. This allows the first level on-chip data cache to achieve a size of 32 KB. To achieve the performance equivalent point of an (8-byte, 32KB) system, a 4-byte system needs to have a cache of 128KB. This 128KB cache will consume too much die area for the on-chip memory and hurt the CPU clock cycle time. For this situation, doubling the bus width is a better choice to achieve the performance level for current technology.

On the other hand, applications like hand-held machines usually require a small CPU packaging size. In embedded applications, CPU packaging cost may be a concern. For these applications, a 4-byte system with a cache of less than 32KB is quite attractive because it achieves the performance level of an 8-byte system with a cache of up to 8KB. For these SPEC92 benchmarks, we observe that doubling the bus trades more die area when the existing cache size is large. This effect is due to the saturation of the hit ratio as the cache size increases.

4. Connecting to the main memory

A processor may be connected directly to the main memory system. The main memory system is usually constant-time dominated, and has a C/β ratio much greater than one. For instance, a 167 MHz x86 system in a typical implementation uses 21 CPU

Table 4. Systems at performance equivalent point for the SPEC92 benchmarks. The constant time is 21 CPU clocks and the transfer time is 6 CPU clock cycles.

Base width = 4 bytes	Base width = 8 bytes
(4-byte, 4KB) = (8-byte, 1KB)	(8-byte, 4KB) = (16-byte, 1-2KB)
(4-byte, 8KB) = (8-byte, 4KB)	(8-byte, 8KB) = (16-byte, 4-8KB)
(4-byte, 16KB) = (8-byte, 8KB)	(8-byte, 16KB) = (16-byte, 8-16KB)
(4-byte, 32KB) = (8-byte, 8-16KB)	(8-byte, 32KB) = (16-byte, 16KB)
(4-byte, 64KB) = (8-byte, 16-32KB)	(8-byte, 64KB) = (16-byte, 16-32KB)
(4-byte, 128KB) = (8-byte, 32-64KB)	(8-byte, 128KB) = (16-byte, 64KB)

clocks for the constant time and 6 CPU cycles for the transfer time. In this case, the 32-bit data path trades about 5% of the hit ratio (ref. Fig. 1), which is about half of the high bound. A larger constant time has an adverse effect on the hit ratio that the bus width can trade as shown in Fig. 1 for the constant time of 11, 21, and 31. This is quite clear because doubling the bus width has a performance contribution only as the per-bus transfer cycles increase. Namely, in a constant time dominated memory system, doubling the bus width may only give a little performance improvement which can be easily obtained by improving other cache structures, such as a higher set associativity. This design space corresponds to the lower left corner in Fig. 1.

We use 21 CPU clocks for the constant time and 6 CPU clocks for the transfer time and examine the tradeoff between doubling the bus width and on-chip data caches. These results are shown in Table 4. The tradeoff shows that in a more constant time dominated memory system, doubling the bus trades a smaller cache size as can be seen by comparing Table 4 and Table 3. In general, when a processor is connected to a memory with a larger C/β ratio, doubling the bus width trades half or less than half of the high bound traded in a transfer time dominated memory system. Doubling the bus width is more advantageous when the processor is designed with the use of a high-speed non-constant-time-dominated L2 cache in mind.

5. Caches and transfer speed

Another alternative to improve the performance is to increase the transfer speed of the bus without changing the bus width. The tradeoff between a cache and bus transfer speed is addressed as follows.

Theorem 2.

If a system using a transfer time of β with a hit ratio of HR_1 and the other using a transfer time of $\beta/2$ with a hit ratio of HR_2 have the same performance, then the hit ratio difference that trades the performance of reducing the transfer clocks by half is between 0 to $1-HR_1$.

Proof:

The CPU execution time for the system with β is

$$E_\beta = (E_i + (E_m - \frac{R}{L})h_t + \frac{R(1+\alpha)}{L}(C + \beta\frac{L}{D}))\times sp \quad (10)$$

while the CPU execution time for the system with 0.5 β is

$$E_{\frac{\beta}{2}} = (E_i + (E_m - \frac{R'}{L})h_t + \frac{R'(1+\alpha')}{L}(C + \frac{\beta}{2}\frac{L}{D}))\times sp . \quad (11)$$

The relationship between Eq. (10) and Eq. (11) is the exact same one as for Eq. (3) and Eq. (4). Thus, hit ratio and reducing the transfer clocks by half hold the same results as the tradeoff between doubling the bus width and the hit ratio.

6. Memory cycles that are hidden from the CPU execution time

Current processors employ many techniques to mitigate the penalty of memory latency and thus improve the IPC (instruction per clock) on overall program performance. For instance, out-of-order execution allows independent instructions to be issued for execution while the load-dependent instructions are buffered until the operand data are available. A non-blocking cache permits multiple misses and does not block the execution of the CPU. Memory system optimizing techniques such as read-bypassing store buffers or data prefetching allow the early initiation of the requests to reduce or hide the read penalty [2].

In Eq. (3), we use a speedup factor for a lumped representation of the case that a portion of the total memory cycles is hidden from the execution of the CPU. To explain the validity of this assumption on the tradeoff model, we use the case when a non-blocking cache is used to hide the memory latency. Assume that when a non-blocking cache is used in the

D-byte system, the portion of cache miss cycles that are not hidden from the CPU execution time is described by

$$\phi \times \frac{R}{L}(C+\beta\frac{L}{D}),\ 0\le\phi\le 1, \tag{12}$$

and the portion of cache miss cycles that are not hidden from the CPU execution time in the 2D-byte system is represented by

$$\phi' \times \frac{R'}{L}(C+\beta\frac{L}{2D}),\ 0\le\phi'\le 1, \tag{13}$$

Specifically, the percentage of hidden memory cycles in the D-byte system is

$$\frac{H_c}{\frac{R}{L}(C+\beta\frac{L}{D})} = 1-\phi \tag{14}$$

and

$$\frac{H'_c}{\frac{R'}{L}(C+\beta\frac{L}{2D})} = 1-\phi' \tag{15}$$

in the 2D-byte system where H_c and H'_c are the number of hidden cache miss cycles for the D-byte and 2D-byte wide system respectively. Because we tradeoff the hit ratio (based on the non-blocking feature) and the bus width, it is necessary for the D-byte and 2D-byte system to have the same percentage of hidden cache miss cycles for a fair comparison. That is, ϕ equals to ϕ'. It is possible to tradeoff the hit ratio of a full blocking cache with the performance of a non-blocking feature if one knows the parameter ϕ for the non-blocking feature. With $\phi=\phi'$, this property is the same as we use the same flush percentage of α and α' in the comparison of hit ratio and bus width. Because ϕ and ϕ' represent the portion of memory cycles that are not hidden from the CPU execution time, they can be used to quantify the effect of other architectural techniques besides the non-blocking feature that can hide the memory latency. These include an out-of-order execution processor model, or data prefetching. From these, we can state the tradeoff results more generally as follows.

Theorem 3.

The tradeoff results between the hit ratio bounds and bus width in Theorem 1 are independent of the use of the mechanisms that are capable of hiding the memory latency.

Proof:

$$\text{For } E_D=E_{2D},\ r=\frac{R'}{R}=\frac{\phi(1+\alpha)(C+\beta\frac{L}{D})-h_t}{\phi'(1+\alpha')(C+\beta\frac{L}{2D})-h_t}.$$

For the high bound, $r=\frac{R'}{R}=2$ given that $\alpha=\alpha'$ and $\phi=\phi'$. For the low bound, $r=\frac{R'}{R}=1$. Therefore, the hit ratio difference of bound that trades the performance of doubling the data path is independent of the use of the memory latency hiding techniques.

V.GENERALIZATION FOR NON-CACHEABLE MEMORY REFERENCES

In this section, we generalize the tradeoff model to include non-cacheable memory references, which typically occur for transferring data in frame buffer regions. A non-cacheable memory reference may require data from a single byte to a size larger than D bytes depending on the load/store instruction executed. For instance, a non-cacheable byte-write makes full use of a D-byte data path even though only one byte is written. On the other hand, a non-cacheable read usually fetches with D-byte data and the CPU extracts the required portion to process if the required data size is less than or equal to D bytes. Larger data size may require multiple D-byte bus cycles. Let N represent the number of non-cacheable load/store instructions executed. In particular, for a D-byte data path,

$$N = N_{1B}+\frac{N_{2B}}{2}+\ldots+\frac{N_{DB}}{2}+\ldots+\frac{N_{jD}}{jD}$$

$$=\sum_{i=1}^{D}\frac{N_{iB}}{i}+\sum_{j=2}^{\frac{w}{D}}\frac{N_{jD}}{jD},\ i=1,\ 2,\ 3,\ \ldots,\ D,\ j=2,\ 4,\ \ldots\frac{w}{D}$$

where w is the maximum operand or instruction size which exceeds the bus width D while N_{iB} (N_{jD}) is the number of data bytes read or written using i ($j\times D$) bytes of the processor data bus. If we compare two systems for data path advantage, the number of non-cacheable load/store instructions executed should be the same for the two systems. Assuming that each non-cacheable cycle takes $C+\beta$ clocks, then for a D-byte data path, it takes

$$(\sum_{i=1}^{D}\frac{N_{iB}}{i}+\sum_{j=2}^{\frac{w}{D}}\frac{N_{jD}}{D})(C+\beta) \tag{16}$$

cycles to execute the N non-cacheable load/store instructions. For a 2D-byte data path, the same N non-cacheable load/store instructions take

$$(\sum_{i=1}^{D}\frac{N_{iB}}{i}+\sum_{j=2}^{\frac{w}{D}}\frac{N_{jD}}{2D})(C+\beta) \tag{17}$$

cycles. The CPU execution time including N non-cacheable load/store instructions for a D-byte data path system is

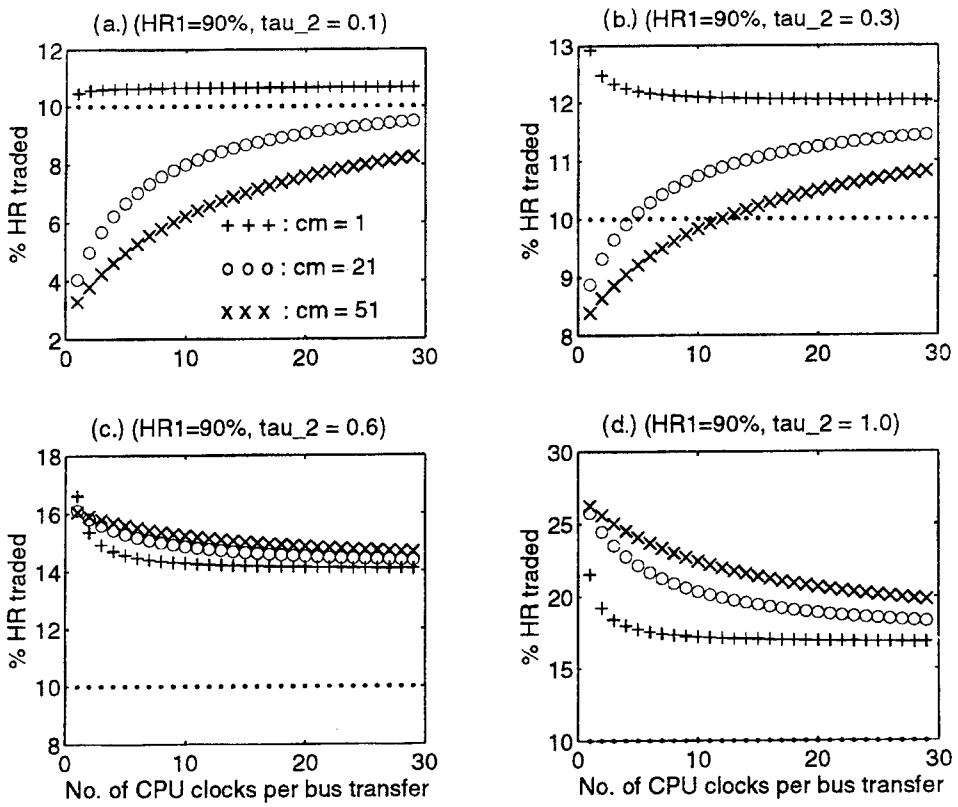


Fig. 3. Effect of Non-Cacheable 2D-byte Memory References on the Tradeoffs Between Doubling the Data Path and Hit Rate (Base Width=4 Bytes).

$$E_{(D,NC)}=(E_D+(\sum_{i=1}^{D}\frac{N_{iB}}{i}+\sum_{j=2}^{\frac{w}{D}}\frac{N_{jD}}{D})(C+\beta))\times sp \qquad (18)$$

while a 2D-byte data path system is

$$E_{(2D,NC)}=(E_{2D}+(\sum_{i=1}^{D}\frac{N_{iB}}{i}+\sum_{j=2}^{\frac{w}{D}}\frac{N_{jD}}{2D})(C+\beta))\times sp \qquad (19)$$

where E_D and E_{2D} are specified as in Eq. (3) and Eq. (4) respectively but without sp. Let $N_{jD}=\tau_j R$. Solving $E_{(D,NC)}=E_{(2D,NC)}$, we obtain

$$r=\frac{R'}{R}=\frac{(1+\alpha)(C+\beta\frac{L}{D})+\frac{L}{D}\Phi(C+\beta)-h_t}{(1+\alpha')(C+\beta\frac{L}{2D})-h_t} \qquad (20)$$

where $\Phi=(\sum_{j=2}^{w/D}\tau_j-\sum_{j=2}^{w/D}\frac{\tau_j}{2})$. As before, HR_1 is the hit ratio associated with the D-byte system while HR_2 is the hit ratio associated with the 2D-byte system. The hit ratio difference that trades the performance of a D-byte width is given by $HR_1-HR_2=(r-1)/(s+1)$ where $s=HR_1/(1-HR_1)$ and $r=R'/R$ as specified by Eq. (20).

Theorem 4.

Considering non-cacheable memory references, if a D-byte system and a 2D-byte system have the same performance, then the hit ratio difference that trades the performance of a D-byte wide data path approaches the limiting value $(1-HR_1)+(4/3)\Phi(1-HR_1)$ for a large per-bus transfer time.

Proof:

Using Eq. (20), $r=R'/R=2+(4/3)\Phi$ for a large transfer time assuming $\alpha=\alpha'=0.5$. Therefore, $HR_1-HR_2=(1-HR_1)+(4/3)\Phi(1-HR_1)$ from Eq. (8).

If $\tau_j=0$, it means that there are no non-cacheable memory references that can use the $j\times D$ wide bus. However, it must be emphasized that if $\tau_j=0$, it does not mean that there are no non-cacheable memory references because a portion of the non-cacheable traffic is specified by $\sum_{i=1}^{D}\frac{N_{iB}}{i}$ which just uses at most a D-byte wide data path for the transfers. For a large transfer time, the non-cacheable 2D-byte traffic has increased the amount of hit ratio traded to exceed the

high bound in Theorem 1 in which all memory references are assumed to be cacheable. Fig. 3 illustrates the effect of non-cacheable 2D-byte traffic on the tradeoff between the data path and the hit rate. The non-cacheable 2D-byte traffic determined by τ_2 is composed of non-cacheable memory references that use a 2D wide bus. In a D-byte system, it would take twice the number of memory cycles to move these data.

In Fig. 3(a), a small percentage of non-cacheable 2D-byte traffic is used to illustrate the hit ratio traded for doubling the data path. Because the non-cacheable 2D-byte traffic is small, the tradeoff curves are similar to those in Fig. 1. As the non-cacheable 2D-byte traffic is increased, the behavior of the tradeoff curves changes as the portion of the $\Phi(C+\beta)$ becomes more weighted. Fig. 3(a) to Fig. 3(d) show the evolution of tradeoff behavior due to the increase in non-cacheable 2D-byte traffic from τ_2=0.1 to τ_2=1.0. In Fig. 3(d) where τ_2=1.0, the curve for constant time = 51 trades more hit ratio than the curve for constant time = 21 or 1. This is totally opposite to the situation in Fig. 3(a) where τ_2=0.1. The reason that the bus width trades more hit ratio is due to $\Phi(C+\beta)$.

Note that doubling the bus width trades a lot of the hit ratio when a large percentage of the non-cacheable 2D-byte memory traffic exists as shown in Fig. 3(c) and Fig. 3(d). If this traffic does not exist, then the tradeoff between doubling the data bus and hit ratio is the same as the case where all memory traffic is assumed to be cacheable. This result is stated as follows.

Theorem 5.

If the non-cacheable memory traffic exists but without 2D-byte non-cacheable memory references, then the hit ratio difference that trades the performance of a D-byte wide data path is between 0 (low bound) and $1-HR_1$ (high bound).

Proof:

Since τ_2=0, the memory reference ratio r in Eq. (20) is the same as in Eq. (5). Therefore, the tradeoff result is the same as in Theorem 1.

VI. MODEL VERIFICATION

The major portion of this section appeared in our previous study [2] where the same tradeoff criterion was used to assess the performance of various hardware techniques. The purpose for including the material here is to provide a better self-content of this paper. The model is verified through the use of the same methodology to find out the performance tradeoffs between line sizes and hit ratio, and compare the results with those previously published. Using our tradeoff model, we will show that the tradeoff approach presented in this paper obtains the exact same results as those of Smith's [10]. In addition, the tradeoff approach can be used to quantify the inter-relationship in line size, cache hit ratio, and memory cycle times.

Given a cache size, a larger cache line size (up to a certain range) usually results in a higher hit ratio than a smaller line size for the same application [10]. We pose the question: how much of a hit (miss) ratio difference between a larger line and a smaller line is necessary to justify the advantage of using a large line size in terms of mean memory delay time. The tradeoff is determined by setting the simplified execution time to be equal, i.e., $E_{FS}=E^*_{FS}$, where

$$E_{FS} = (E - \frac{R}{L_0}) + (\frac{R(1+\alpha)}{L_0})(C + \beta\frac{L_0}{D}) \tag{21}$$

and

$$E^*_{FS} = (E - \frac{R^*}{L^*}) + (\frac{R^*(1+\alpha^*)}{L^*})(C + \beta\frac{L^*}{D}) \tag{22}$$

and obtain

$$\frac{R^*}{R} = (\frac{(1+\alpha)(C+\beta\frac{L_0}{D})-1}{(1+\alpha^*)(C+\beta\frac{L^*}{D})-1})(\frac{L^*}{L_0}). \tag{23}$$

In E_{FS} and E^*_{FS}, the speed up factor, sp, is not shown for clarity. Let EHR be the hit ratio for using a larger line size L^*, and HR is the hit ratio for using a smaller line size L_0. Then, the hit ratio difference for the equivalence of mean memory delay time is

$$\Delta EHR_{L^*} = EHR - HR = MR - EMR = \frac{1-r}{s+1} \tag{24}$$

where $s=HR/(1-HR)$ and $r=(R^*/L^*)/(R/L_0)$. ΔEHR_{L^*} is the minimum hit (miss) ratio difference required for using a larger line size L^* to have the same performance as using a smaller line size L_0. Then, we use this tradeoff relationship and combine with Smith's approach to determine the optimal line size [10]. An optimal line size is determined by finding the least average memory delay per memory reference. Suppose that we want to determine the optimal line size from the set of y line sizes represented by $\{L_i | 1 \le i \le y\}$. For $1 \le i \le y$, the optimal line size is determined by the following minimum operations.

$$\text{Min}\{(1-HR_{Li})(C+\beta\frac{L_i}{D}) + HR_{Li}\}. \tag{25}$$

The hit cycle time is one cycle here. We use the line

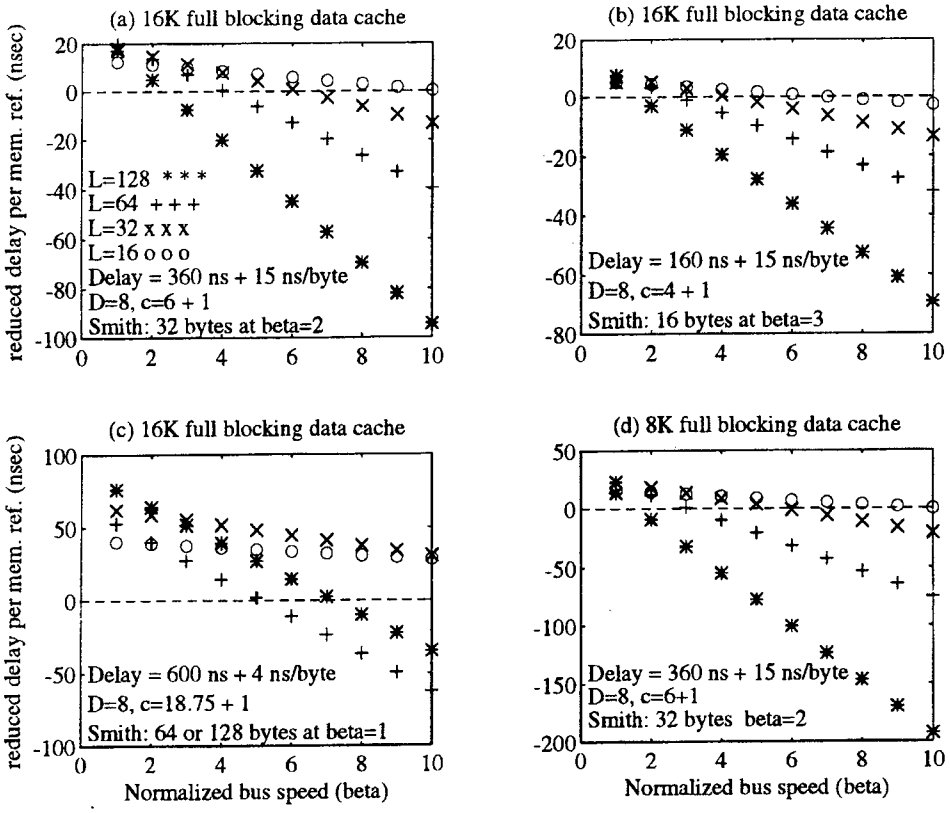


Fig. 4. Validation with Smith's Design Target Hit Ratio for Data Caches

size L_0 as a base case for the comparison of mean memory delay with the set of y line sizes represented by $\{L_i | 1 \le i \le y\}$ where $L_i > L_0$. In this setting, we can examine whether a larger line size offers a performance advantage or not due to its higher hit ratio. Let the hit ratio of line L_i be denoted as HR_{Li} and HR_{L0} for L_0. We consider the range of line sizes when $HR_{Li} \ge HR_{L0}$. Based on the minimum mean memory delay approach, the best line is determined by the following equivalent maximum operations.

$$\begin{aligned} &\mathrm{Max}\{((1-HR_{L0})-(1-HR_{Li})) \\ &(C+\beta\frac{L_i}{D})+HR_{L0}-HR_{Li}\} \\ &=\mathrm{Max}\{(\Delta HR_{Li}(C-1+\beta\frac{L_i}{D})\}. \end{aligned} \tag{26}$$

We use the largest difference of the mean memory delay time between line size L_0 and each of the other line sizes respectively to determine the optimal line size. The largest difference means the smallest mean memory delay of the corresponding line size L_i. The following maximum operation determines the optimal line size and indicates the beneficial range of bus speed or memory access time for using that line size.

$$\mathrm{Max}\{(\Delta HR_{Li}-\Delta EHR_{Li})(C-1+\beta\frac{L_i}{D})\} \tag{27}$$

where ΔEHR_{Li} is specified by Eq. (24) replacing L^* with L_i for $1 \le i \le y$. Line size L_i is justified by its sufficient higher hit ratio being a large size when the above maximum has a value greater than zero. We compare the optimal line size determined by Eq. (27) with the results of Smith's work [10]. They are presented in Fig. 4. Consider Fig. 4(a), for instance. Given 360 ns + 15 ns/byte for the delay time and bus width D=8 bytes, normalizing with 60 ns (a chosen processor cycle time), we obtain C=6+1 and β=(8×15)/60=2. At β=2, the optimal line size determined by Eq. (27) for a 16KB data cache is 32 bytes which has the maximum reduced delay time per memory reference (See Fig. 4(a).) This result is exactly the same one as in Smith's work. The order of line size to choose is 32, 64, 16, and 128

bytes, which also matches Smith's work. Therefore, the performance tradeoff methodology is verified.

VII. CONCLUSION

In this paper, we have examined the performance and cost tradeoffs among cache hit ratio, data path width, and transfer memory design. We have shown quantitatively how to use data path, caches, and memory system design to establish an equivalent performance level where a smaller cost or less design complexity configuration can be chosen for the implementation, based on the available technology. In particular, if a D-byte system and a 2D-byte system have the same performance, then the hit ratio difference that trades the performance of a D-byte wide data path is between 0 (low bound) and 1-*HR* (high bound) where *HR* is the hit ratio associated with the D-byte system. The hit ratio bounds traded for the D-byte data path are independent of the use of memory latency hiding mechanisms. The constant time and the per-bus transfer time of the memory system are used to determine the exact amount of hit ratio traded. In general, when a processor is connected to a memory with a large C/β ratio, doubling the bus width trades half or less than half of the high bound traded in a transfer time dominated memory system. For current main memory systems, doubling the data path trades about half of the high bound traded in a transfer-time dominated system. Doubling the bus width trades a lot of the hit ratio when a large percentage of non-cacheable 2D-byte memory traffic exists. A designer should carefully evaluate the characteristics of the applications for the amount of the non-cacheable 2D-byte memory traffic when considering bus width or cache improvement. If the 2D-byte traffic does not exist or is very small in amount, then the tradeoff between doubling the data bus and hit ratio is the same as the case where all memory traffic is assumed to be cacheable.

Applications like hand-held machines usually require a small CPU packaging size. In embedded applications, CPU packaging costs may be a concern. For these applications, a 4-byte system with a cache of less than 32KB is quite attractive because it achieves the performance level of an 8-byte system with a cache of up to 8KB. The performance of doubling the bus width trades a hit ratio close to the high bound if the processor is connected to a transfer time dominated memory system. For this, doubling the data bus is more advantageous when the processor is designed to be used with a high-speed non-constant-time-dominated L2 cache. For the SPEC92 benchmarks, we observe that doubling the bus trades more die area when the existing cache size is large.

In a constant time dominated system, the effectiveness of doubling the data path is limited if no 2D-byte non-cacheable memory references exist. In systems using a burst transfer memory design, the attempt to double the data path should be accompanied by a reduction of the constant time. Otherwise, a cache structure enhancement, such as a higher set associativity, can easily achieve the performance improvement of doubling the data path. However, if a large amount of non-cacheable 2D-byte memory traffic exists, doubling the bus width should be considered first rather than improving the cache size. In this paper, we have explored the design space among the hit ratio, data path width, and transfer memories through a performance tradeoff methodology. The approach is further generalized to include non-cacheable memory references so that it can be used to explore the design space for general cases.

ACKNOWLEDGEMENTS

The author would like to thank the reviewers for their helpful comments on the paper. The NSC in part supports this research (NSC 85-2213-E-224-021).

REFERENCES

1. Bugge, H.O. E.H. Kristiansen, and B.O. Bakka, "Trace-driven Simulations for a Two-Level Cache Design in Open-Bus Systems," *Proceedings of the 17th International Symposium on Computer Architecture*, pp. 250-259, June 1990.
2. Chen C.-H. and A.K. Somani,"Architecture Technique Tradeoffs Using Mean Memory Delay Time," *IEEE Transactions on Computers*, Vol. 45, No. 10, October 1996.
3. Gee, J.D. M.D. HIll, D.N. Pnevmatikatos, and A. J. Smith, "Cache Performance of the SPEC92 Benchmark Suite," *IEEE Micro*, pp. 17-27, August 1993.
4. Hennessy J.L. and D.A. Patterson, Computer Architecture, A Quantitative Approach, Morgan Kaufmann Publishers, Inc., 1996.
5. Johnson R.E. and F.L. Kiokemeister, Calculus with Analytic Geometry, Allyn and Bacon Inc., pp. 393-395, 1978.
6. Jouppi N.P. and S.J.E. Wilton, "Tradeoffs in Two-Level On-Chip Caching," *Proceedings of the 21st International Symposium on Computer Architecture*, pp. 34-45, 1994.
7. Mulder, J.M. N.T. Quach, and M.J. Flynn, "An Area Model for On-Chip Memories and its Application," *IEEE Journal of Solid-State Circuits*, Vol. 26, No. 2, February 1991.
8. Przybylski, S.M. Horowitz, and J. Hennessy, "Performance Tradeoffs in Cache Design," *Proceedings of the 15th International Symposium on*

Computer Architecture, pp. 290-298, May 1988.
9. Smith, A.J. "Cache Evaluation and Impact of Workload Choice," *Proceedings of the 12th International Symposium on Computer Architecture*, pp. 64-73, June 1985.
10. Smith, A.J. "Line (Block) Size Choice for CPU Cache Memories," *IEEE Transactions on Computers*, Vol. C-36, No. 9, pp. 1063-1075, September 1987.
11. Sohi G.S. and M. Franklin, "High-Bandwidth Data Memory Systems for Superscalar Processors," *Proceedings of the 4th International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 53-62, 1991.
12. Wada, T.S. Rajan, and S.A. Przybylske, "An Analytical Access Time Model for On-Chip Cache Memories," *IEEE Journal of Solid-State Circuits*, Vol. 27, No. 8, August 1992.
13. Yeager, K.C. "The Mips R10000 Superscalar Microprocessor," *IEEE Micro*, pp. 28-40, April 1996.

Discussions of this paper may appear in the discussion section of a future issue. All discussions should be submitted to the Editor-in-Chief.

Manuscript Received: June 25, 1997
Revision Received: Dec. 09,1997
and Accepted: Jan. 24, 1998

快取記憶體，資料匯流排寬度，及爆發式傳送記憶體設計空間之探討

陳中和

國立雲林科技大學電子工程技術系

摘 要

快取記憶體，資料匯流排寬度，爆發式傳送記憶體三者爲降低處理器與主記憶體資料傳遞時間之主要硬體技術。本文以一評量方法探討快取記憶體命中率，資料匯流排寬度，及爆發式傳送記憶體三者之效能關係。我們的結果顯示如果一個資料匯流排爲D位元組寬的系統與一資料匯流排寬爲2D位元組的系統有同樣的效能，則換取資料匯流排D位元組寬之效能之快取記憶體命中率介於0（低限）到*1-HR*（高限）之間。其中*HR*爲資料匯流排爲D位元組寬的系統之快取記憶體命中率。就目前受傳送時間主宰之主記憶體而言，加倍資料匯流排寬度之效益大約換取上述高限的一半。加倍資料匯流排寬度之好處以微處理器搭配高速之L2快取記憶體較爲有利。若系統有相當大量之非快取記憶體存取動作，則加倍資料匯流排寬度可換取相當大之快取記憶體命中率。

關鍵字：快取記憶體，資料匯流排寬度，爆發式傳送記憶體，效益評量。

GENERALIZED SOURCE CODING THEOREMS AND HYPOTHESIS TESTING: PART I -- INFORMATION MEASURES

Po-Ning Chen*
Dept. of Communications Engineering
National Chiao Tung University
Hsin Chu, Taiwan 300, R.O.C.

Fady Alajaji
Dept. of Mathematics and Statistics
Queen's University
Kingston, Ontario K7L 3N6, Canada

Key Words: information theory, entropy, mutual information, divergence,e-capacity.

ABSTRACT

Expressions for ε-entropy rate, ε-mutual information rate and ε-divergence rate are introduced. These quantities, which consist of the quantiles of the asymptotic information spectra, generalize the inf/sup-entropy/information/divergence rates of Han and Verdú. The algebraic properties of these information measures are rigorously analyzed, and examples illustrating their use in the computation of the ε-capacity are presented. In Part II of this work, these measures are employed to prove general source coding theorems for block codes, and the general formula of the Neyman-Pearson hypothesis testing type-II error exponent subject to upper bounds on the type-I error probability.

I. INTRODUCTION AND MOTIVATION

Entropy, divergence and mutual information are without a doubt the most important information theoretic quantities. They constitute the fundamental measures upon which information theory is founded. Given a discrete random variable X with distribution P_X, its entropy is defined by [7]

$$H(X) \stackrel{\Delta}{=} -\sum_x P_X(x)\log_2 P_X(x) = E_{P_X}[-\log P_X(x)].$$

$H(X)$ is a measure of the average amount of uncertainty in X. The divergence, on the other hand, measures the relative distance between the distributions of two random variables X and $\hat{X}$ that are defined on the same alphabet :

$$D(X\|\hat{X}) \stackrel{\Delta}{=} E_{P_X}[\log_2 \frac{P_X(x)}{P_{\hat{X}}(X)}].$$

As for the mutual information $I(X;Y)$ between random variables X and Y, it represents the average amount of information that Y contains about X. It is defined as the divergence between the joint distribution P_{XY} and the product distribution P_XP_Y:

$$I(X;Y) \stackrel{\Delta}{=} D(P_{XY}\|P_XP_Y) = E_{P_{XY}}[\log_2 \frac{P_{XY}(X,Y)}{P_X(X)P_Y(Y)}].$$

More generally, consider an input process $\boldsymbol{X}$ defined by a sequence of finite dimensional distributions [11]: $\boldsymbol{X} \stackrel{\Delta}{=} \{X^n = (X_1^{(n)}, ..., X_n^{(n)})\}_{n=1}^{\infty}$. Let $\boldsymbol{Y} \stackrel{\Delta}{=} \{Y^n = (Y_1^{(n)}, ..., Y_n^{(n)})\}_{n=1}^{\infty}$ be the corresponding output process induced by $\boldsymbol{X}$ via the channel $\boldsymbol{W} \stackrel{\Delta}{=} \{W^n = (W_1^{(n)}, ..., W_n^{(n)})\}_{n=1}^{\infty}$, which is an arbitrary sequence of n-dimensional conditional distributions from $\mathcal{X}^n$ to $\mathcal{Y}^n$, where $\mathcal{X}$ and Υ are the input and output alphabets respectively. The entropy rate for the source $\boldsymbol{X}^n$ is defined by[2], [7]

*Correspondence addressee

$$H(X) \triangleq \lim_{n\to\infty} \frac{1}{n} E[-\log P_{X^n}(X^n)],$$

assuming the limit exists. Similarly the expressions for the divergence and mutual information rates are given by

$$D(X \| \hat{X}) \triangleq \lim_{n\to\infty} \frac{1}{n} E[\log \frac{P_{X^n}(X^n)}{P_{\hat{X}^n}(X^n)}],$$

and

$$I(X;Y) \triangleq \lim_{n\to\infty} \frac{1}{n} E[\log \frac{P_{X^nY^n}(X^n,Y^n)}{P_{X^n}(X^n)P_{Y^n}(Y^n)}],$$

respectively.

The above quantities have an operational significance established via Shannon's coding theorems when the stochastic systems under consideration satisfy certain *regularity* conditions (such as stationarity and ergodicity, or information stability) [9], [11]. However, in more complicated situations such as when the systems are non-stationary (with time-varying statistics), these information rates are no longer valid and lose their operational significance. This results in the need to establish new information measures which appropriately characterize the operational limits of arbitrary stochastic systems.

This is achieved in [10] and [11] where Han and Verdú introduce the notions of *inf/sup-entropy/information rates* and illustrate the key role these information measures play in proving a general lossless (block) source coding theorem and a general channel coding theorem. More specifically, they demonstrate that for an arbitrary finite-alphabet source $\boldsymbol{X}$, the expression for the minimum achievable (block) source coding rate is given by the *sup-entropy rate* $\overline{H}(\boldsymbol{X})$, defined as the *limsup in probability* of $(1/n) \log 1/P_{X^n}(X^n)$ [10]. They also establish in [11] the formulas of the ε-capacity C_ε and capacity[1] C of arbitrary single-user channels without feedback (not necessarily information stable, stationary, ergodic, etc.). More specifically, they show that

$$\sup_X \sup \{R: F_X(R) < \varepsilon\} \le C_\varepsilon \le \sup_X \sup \{R: F_X(R) \le \varepsilon\},$$

and

$$C = \sup_X \underline{I}(X;Y),$$

where

$$F_X(R) \triangleq \limsup_{n\to\infty} \Pr[(1/n) i_{X^nY^n}(X^n;Y^n) \le R],$$

$(1/n) i_{X^nY^n}(X^n; Y^n)$ is the sequence of normalized information densities defined by

$$i_{X^nY^n}(X^n;Y^n) = \log \frac{P_{Y^n|X^n}(y^n \mid x^n)}{P_{Y^n}(y^n)},$$

and $\underline{I}(X;Y)$ is *inf-information rate* between $\boldsymbol{X}$ and $\boldsymbol{Y}$, which is defined as the *liminf in probability* of $(1/n) i_{X^nY^n}(X^n;Y^n)$.

By adopting the same technique as in [10] (also in [11]), general expressions for the capacity of single-user channels with feedback and for Neyman-Pearson type-II error exponents are derived in [4] and [6], respectively. Furthermore, an application of the type-II error exponent formula to the non-feedback and feedback channel reliability functions is demonstrated in [6] and [5].

The above inf/sup-entropy/information rates are expressed in terms of the *liminf/limsup in probability* of the normalized entropy/information densities. The *liminf in probability* of a sequence of random variables is defined as follows [10]: if A_n is a sequence of random variables, then its *liminf in probability* is the largest extended real number $\underline{U}$ such that for all $\xi>0$,

$$\lim_{n\to\infty} \Pr[A_n \le \underline{U} - \xi] = 0. \quad (1)$$

Similarly, its *limsup in probability* is the smallest extended real number $\overline{U}$ such that for all $\xi>0$,

$$\lim_{n\to\infty} \Pr[A_n \ge \overline{U} + \xi] = 0. \quad (2)$$

Note that these two quantities are always defined; if they are equal, then the sequence of random variables converges in probability to a constant.

It is straightforward to deduce that Eqs. (1) and (2) are respectively equivalent to

$$\liminf_{n\to\infty} \Pr[A_n \le \underline{U} - \xi] = \limsup_{n\to\infty} \Pr[A_n \le \underline{U} - \xi] = 0. \quad (3)$$

and

$$\liminf_{n\to\infty} \Pr[A_n \ge \overline{U} + \xi] = \limsup_{n\to\infty} \Pr[A_n \ge \overline{U} + \xi] = 0. \quad (4)$$

[1] *Definition* ([8], [11]): Given $0<\varepsilon<1$, an (n, M, ε) code for the channel $\boldsymbol{W}$ has blockength n, M codewords and average (decoding) error probability not larger than ε. A non-negative number R is an *ε-achievable rate* if for every $\delta>0$, there exist, for all n sufficiently large, (n, M, ε) codes with rate $(1/n) \log M > R-\delta$. The *supremum* of all ε-achievable rates is called the *ε-capacity*, C_ε. The *capacity* C is the supremum of rates that are ε-achievable for all $0<\varepsilon<1$ and hence $C=\lim_{\varepsilon\downarrow 0} C_\varepsilon$.

In other words, C_ε is the largest rate at which information can be conveyed over the channel such that the probability of a decoding error is below a fixed threshold ε, for sufficiently large blocklengths, Furthermore, C represents the largest rate at which information can be transmitted over the channel with asymptotically *vanishing* error probability.

We can observe however that there might exist cases of interest where *only* the liminfs of the probabilities in (3) and (4) are equal to zero; while the limsups do *not* vanish. There are also other cases where *both* the liminfs and limsups in (3)-(4) do not vanish; but they are upper bounded by a prescribed threshold. Furthermore, there are situations where the interval $[\underline{U}, \overline{U}]$ does not contain only one point; for e.g., when A_n converges in distribution to another random variable. Hence, those points within the interval $[\underline{U}, \overline{U}]$ might possess a Shannon-theoretic operational meaning when for example A_n consists of the normalized entropy density of a given source.

The above remarks constitute the motivation for this work in which we generalize Han and Verdú's information rates and prove general data compression and hypothesis testing theorems that are the *counterparts* of their ε-capacity channel coding theorem [11].

In Part I, we propose generalized versions of the inf/sup-entropy/information/divergence rates. We analyze in detail the algebraic properties of these information measures, and we illustrate their use in the computation of the ε-capacity of arbitrary additive-noise channels. In Part II of this paper [3], we utilize these quantities to establish general source coding theorems for arbitrary finite-alphabet sources, and the general expression of the Neyman-Pearson type-II error exponent.

II. GENERALIZED INFORMATION MEASURES

Definition 1. (Inf/sup-spectrum)

If $\{A_n\}_{n=1}^{\infty}$ is a sequence of random variables, then its *inf-spectrum* $\underline{u}(\cdot)$ and its *sup-spectrum* $\overline{u}(\cdot)$ are defined by

$$\underline{u}(\theta) \triangleq \liminf_{n\to\infty} \Pr[A_n \le \theta],$$

and

$$\overline{u}(\theta) \triangleq \limsup_{n\to\infty} \Pr[A_n \le \theta].$$

In other words, $\underline{u}(\cdot)$ and $\overline{u}(\cdot)$ are respectively the liminf and the limsup of the cumulative distribution function (CDF) of A_n. Note that by definition, the CDF of A_n – $\Pr\{A_n \le \theta\}$–is non-decreasing and right-continuous. However, for $\underline{u}(\cdot)$ and $\overline{u}(\cdot)$, only the *non-decreasing* property remains[2].

Definition 2. (Quantile of inf/sup-spectrum)

For any $0 \le \delta \le 1$, the quantiles $\underline{U}_\delta$ and $\overline{U}_\delta$ of the sup-spectrum and the inf-spectrum are defined by[3]

$$\underline{U}_\delta \triangleq \begin{cases} -\infty & \text{if } \{\theta: \overline{u}(\theta) \le \delta\} = \varnothing, \\ \sup\{\theta: \overline{u}(\theta) \le \delta\}, & \text{otherwise,} \end{cases}$$

and

$$\overline{U}_\delta \triangleq \begin{cases} -\infty & \text{if } \{\theta: \underline{u}(\theta) \le \delta\} = \varnothing, \\ \sup\{\theta: \underline{u}(\theta) \le \delta\}, & \text{otherwise,} \end{cases}$$

respectively. It follows from the above definitions that $\underline{U}_\delta$ and $\overline{U}_\delta$ are right-continuous and non-decreasing in δ.

Note that the liminf in probability $\underline{U}$ and the limsup in probability $\overline{U}$ of A_n satisfy

$$\underline{U} = \underline{U}_0$$

and

$$\overline{U} = \overline{U}_{1^-},$$

respectively, where the superscript "–" denotes a strict inequality in the definition of $\overline{U}_{1^-}$; i.e.,

$$\overline{U}_{\delta^-} \triangleq \sup\{\theta: \underline{u}(\theta) < \delta\}.$$

Note also that

$$\underline{U} \le \underline{U}_\delta \le \overline{U}_\delta \le \overline{U}.$$

Remark that $\underline{U}_\delta$ and $\overline{U}_\delta$ always exist. Furthermore, if $\underline{U}_\delta = \overline{U}_\delta \ \forall \delta \in [0,1]$, then the sequence of random variables A_n converges in distribution to a random variable A, provided the distribution sequence of A_n is tight.

For a better understanding of the quantities defined above, we depict them in Fig. 1.

In the above definitions, if we let the random variable A_n equal the normalized entropy density of an arbitrary source X, we obtain two generalized entropy measures for X: *the δ-inf-entropy-rate* $\underline{H}_\delta(X)$

[2] It is pertinent to also point out that even if we do not require right-continuity as a fundamental property of a CDF, the spectrums $\underline{u}(\cdot)$ and $\overline{u}(\cdot)$ are *not* necessarily legitimate CDFs of (conventional real-valued) random variables since there might exist cases where the "probability mass escapes to infinity" (cf. [1, page 346]). A necessary and sufficient condition for $\underline{u}(\cdot)$ and $\overline{u}(\cdot)$ to be conventional CDFs (without requiring right-continuity) is that the sequence of distribution functions of A_n be *tight* [1, page 346]. Tightness is actually guaranteed if the alphabet of A_n is finite.

[3] Note that the usual definition of the quantile function $\phi(\delta)$ of a non-decreasing function $F(\cdot)$ is slightly different from our definition [1, page 190]: $\phi(\delta)=\sup\{\theta: F(\theta)<\delta\}$. Remark that if $F(\cdot)$ is strictly increasing, then the quantile is nothing but the inverse of $F(\cdot)$: $\phi(\delta)=F^{-1}(\delta)$.

and the *δ-sup-entropy-rate* $\overline{H}_\delta(X)$ as described in Table 1. Note that the *inf-entropy-rate* $\underline{H}(X)$ and the *sup-entropy-rate* $\overline{H}(X)$ introduced in [10] are special cases of the δ-inf/sup-entropy rate measures:

$$\underline{H}(X)=\underline{H}_0(X), \text{ and } \overline{H}(X)=\overline{H}_1(X).$$

Analogously, for an arbitrary channel $W\overset{\Delta}{=}P_{Y|X}$ with input X and output Y (or respectively for two observations X and $\hat{X}$), if we replace A_n by the normalized information density (resp. by the normalized log-likelihood ratio), we get the δ-inf/sup-information rates (resp. δ-inf/sup-divergences rates) as shown in Table 1.

The algebraic properties of these newly defined information measures are investigated in the next section.

III. PROPERTIES OF THE GENERALIZED INFORMATION MEASURES

Lemma 1.

Consider two arbitrary random sequences, $\{A_n\}_{n=1}^{\infty}$ and $\{B_n\}_{n=1}^{\infty}$. Let $\overline{u}(\cdot)$ and $\underline{u}(\cdot)$ denote respectively the sup-spectrum and inf-spectrum of $\{A_n\}_{n=1}^{\infty}$. Similarly, let $\overline{v}(\cdot)$ and $\underline{v}(\cdot)$ denote respectively the sup-spectrum and inf-spectrum of $\{B_n\}_{n=1}^{\infty}$. Define $\underline{U}_\delta\overset{\Delta}{=}\sup\{\theta:\overline{u}(\theta)\le\delta\}$, $\overline{U}_\delta\overset{\Delta}{=}\sup\{\theta:\underline{u}(\theta)\le\delta\}$, $\underline{V}_\delta=\sup\{\theta:\overline{v}(\theta)\le\delta\}$, $\overline{V}_\delta\overset{\Delta}{=}\sup\{\theta:\underline{v}(\theta)\le\delta\}$,

$$(\underline{U+V})_{\delta+\gamma}\overset{\Delta}{=}\sup\{\theta:\overline{(u+v)}(\theta)\le\delta+\gamma\},$$

$$\overline{(U+V)}_{\delta+\gamma}\overset{\Delta}{=}\sup\{\theta:(\underline{u+v})(\theta)\le\delta+\gamma\},$$

$$\overline{(u+v)}(\theta)\overset{\Delta}{=}\limsup_{n\to\infty}\Pr\{A_n+B_n\le\theta\},$$

and

$$(\underline{u+v})(\theta)\overset{\Delta}{=}\liminf_{n\to\infty}\Pr\{A_n+B_n\le\theta\},$$

Then the following statements hold.

1. $\underline{U}_\delta$ and $\overline{U}_\delta$ are both non-decreasing functions of $\delta\in[0,1]$.
2. For $\delta\ge0$, $\gamma\ge0$, and $1\ge\delta+\gamma$,

$$(\underline{U+V})_{\delta+\gamma}\ge\underline{U}_\delta+\underline{V}_\gamma, \tag{5}$$

and

$$\overline{(U+V)}_{\delta+\gamma}\ge\underline{U}_\delta+\overline{V}_\gamma. \tag{6}$$

3. For $\delta\ge0$, $\gamma\ge0$, and $1>\delta+\gamma$,

$$(\underline{U+V})_\delta\le\underline{U}_{\delta+\gamma}+\overline{V}_{(1-\gamma)^-}, \tag{7}$$

and

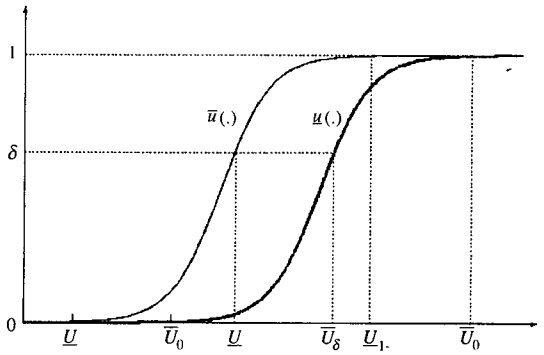


Fig. 1. The asymptotic CDFs of a sequence of random variables $\{A_n\}_{n=1}^{\infty}$. $\overline{u}(\cdot)$=sup-spectrum of A_n; $\underline{u}(\cdot)$=inf-spectrum of A_n.

$$\overline{(U+V)}_\delta\le\overline{U}_{\delta+\gamma}+\overline{V}_{(1-\gamma)^-}. \tag{8}$$

Proof:

The proof of property 1 follows directly from the definitions of $\underline{U}_\delta$ and $\overline{U}_\delta$ and the fact that the inf-spectrum and the sup-spectrum are non-decreasing in δ.

To show (5), we first observe that

$$\Pr\{A_n+B_n\le\underline{U}_\delta+\underline{V}_\gamma\}\le\Pr\{A_n\le\underline{U}_\delta\}+\Pr\{B_n\le\underline{V}_\gamma\}.$$

Then

$$\begin{aligned}&\limsup_{n\to\infty}\Pr\{A_n+B_n\le\underline{U}_\delta+\underline{V}_\gamma\}\\&\le\limsup_{n\to\infty}(\Pr\{A_n\le\underline{U}_\delta\}+\Pr\{B_n\le\underline{V}_\gamma\})\\&\le\limsup_{n\to\infty}\Pr\{A_n\le\underline{U}_\delta\}+\limsup_{n\to\infty}\Pr\{B_n\le\underline{V}_\gamma\}\\&\le\delta+\gamma,\end{aligned}$$

which, by definition of $(\underline{U+V})_{\delta+\gamma}$, yields (5).

Similarly, we have

$$\Pr\{A_n+B_n\le\underline{U}_\delta+\overline{V}_\gamma\}\le\Pr\{A_n\le\underline{U}_\delta\}+\Pr\{B_n\le\overline{V}_\gamma\}.$$

Then

$$\begin{aligned}&\liminf_{n\to\infty}\Pr\{A_n+B_n\le\underline{U}_\delta+\overline{V}_\gamma\}\\&\le\liminf_{n\to\infty}(\Pr\{A_n\le\underline{U}_\delta\}+\Pr\{B_n\le\overline{V}_\gamma\})\\&\le\limsup_{n\to\infty}\Pr\{A_n\le\underline{U}_\delta\}+\liminf_{n\to\infty}\Pr\{B_n\le\overline{V}_\gamma\}\\&\le\delta+\gamma,\end{aligned}$$

which, by definition of $\overline{(U+V)}_{\delta+\gamma}$, proves (6).

To show (7), we remark from (5) that $(\underline{U+V})_\delta+(\underline{-V})_\gamma\le(\underline{U+V-V})_{\delta+\gamma}=\underline{U}_{\delta+\gamma}$. Hence,

Table 1. Generalized information measures where $\delta \in [0,1]$.

Entropy Measures	
System	Arbitrary Source $\boldsymbol{X}$
A_n: Norm. Entropy Density	$(1/n)\, h_{X^n}(X^n) \triangleq -(1/n)\log P_{X^n}(X^n)$
Entropy Sup-Spectrum	$\overline{h}_X(\theta) \triangleq \limsup_{n\to\infty} \Pr\{(1/n)\, h_{X^n}(X^n) \le \theta\}$
Entropy Inf-Spectrum	$\underline{h}_X(\theta) \triangleq \liminf_{n\to\infty} \Pr\{(1/n)\, h_{X^n}(X^n) \le \theta\}$
δ-Inf-Entropy Rate	$\underline{H}_\delta(\boldsymbol{X}) \triangleq \sup\{\theta: \overline{h}_X(\theta) \le \delta\}$
δ-Sup-Entropy Rate	$\overline{H}_\delta(\boldsymbol{X}) \triangleq \sup\{\theta: \underline{h}_X(\theta) \le \delta\}$
Sup-Entropy Rate	$\overline{H}(\boldsymbol{X}) \triangleq \overline{H}_1(\boldsymbol{X})$
Inf-Entropy Rate	$\underline{H}(\boldsymbol{X}) \triangleq \underline{H}_0(\boldsymbol{X})$

Mutual Information Measures	
System	Arbitrary channel $\boldsymbol{W} \triangleq P_{Y\|X}$ with input $\boldsymbol{X}$ and output $\boldsymbol{Y}$
A_n: Norm. Information Density	$(1/n)\, i_{(X^n,Y^n)}(X^n;Y^n) \triangleq (1/n)\log \dfrac{dP_{X^nY^n}}{d(P_{X^n}\times P_{Y^n})}(X^n,Y^n)$
Information Sup-Spectrum	$\overline{i}_{(X,Y)}(\theta) \triangleq \limsup_{n\to\infty} \Pr\{(1/n)\, i_{(X^n,Y^n)}(X^n;Y^n) \le \theta\}$
Information Inf-Spectrum	$\underline{i}_{(X,Y)}(\theta) \triangleq \liminf_{n\to\infty} \Pr\{(1/n)\, i_{(X^n,Y^n)}(X^n;Y^n) \le \theta\}$
δ-Inf-Information Rate	$\underline{I}_\delta(\boldsymbol{X};\boldsymbol{Y}) \triangleq \sup\{\theta: \overline{i}_{(X,Y)}(\theta) \le \delta\}$
δ-Sup-Information Rate	$\overline{I}_\delta(\boldsymbol{X};\boldsymbol{Y}) \triangleq \sup\{\theta: \underline{i}_{(X,Y)}(\theta) \le \delta\}$
Sup-Information Rate	$\overline{I}(\boldsymbol{X};\boldsymbol{Y}) \triangleq \overline{I}_1(\boldsymbol{X};\boldsymbol{Y})$
Inf-Information Rate	$\underline{I}(\boldsymbol{X};\boldsymbol{Y}) \triangleq \underline{I}_0(\boldsymbol{X};\boldsymbol{Y})$

Divergence Measures	
System	Arbitrary sources $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}$
A_n: Norm Log-Likelihood Ratio	$(1/n)\, d_{X^n}(X^n \,\|\, \hat{X}^n) \triangleq (1/n)\log[\, dP_{X^n}/dP_{\hat{X}^n}](X^n)$
Divergence Sup-Spectrum	$\overline{d}_{X\|\hat{X}}(\theta) \triangleq \limsup_{n\to\infty} \Pr\{(1/n)\, d_{X^n}(X^n \,\|\, \hat{X}^n) \le \theta\}$
Divergence Inf-Spectrum	$\underline{d}_{X\|\hat{X}}(\theta) \triangleq \liminf_{n\to\infty} \Pr\{(1/n)\, d_{X^n}(X^n \,\|\, \hat{X}^n) \le \theta\}$
δ-Inf-Divergence Rate	$\underline{D}_\delta(\boldsymbol{X} \,\|\, \hat{\boldsymbol{X}}) \triangleq \sup\{\theta: \overline{d}_{X\|\hat{X}}(\theta) \le \delta\}$
δ-Sup-Divergence Rate	$\overline{D}_\delta(\boldsymbol{X} \,\|\, \hat{\boldsymbol{X}}) \triangleq \sup\{\theta: \underline{d}_{X\|\hat{X}}(\theta) \le \delta\}$
Sup-Divergence Rate	$\overline{D}(\boldsymbol{X} \,\|\, \hat{\boldsymbol{X}}) \triangleq \overline{D}_1(\boldsymbol{X} \,\|\, \hat{\boldsymbol{X}})$
Inf-Divergence Rate	$\underline{D}(\boldsymbol{X} \,\|\, \hat{\boldsymbol{X}}) \triangleq \underline{D}_0(\boldsymbol{X} \,\|\, \hat{\boldsymbol{X}})$

$$(\underline{U+V})_\delta \le \underline{U}_{\delta+\gamma} - (\underline{-V})_\gamma.$$

(Note that the cases $\delta+\gamma=1$ or $\gamma=1$ are not allowed here because they result in $\underline{U}_1=-\underline{V}_1=\infty$, and the subtraction of two infinite terms is undefined. That is why the condition for property 2, $1\le\delta+\gamma$, is replaced by $1>\delta+\gamma$ in property 3.)

The proof is completed by showing that

$$-(\underline{-V})_\gamma \le \overline{V}_{(1-\gamma)^-}. \tag{9}$$

By definition,

$$(\overline{-\nu})(\theta) \triangleq \limsup_{n\to\infty} \Pr\{-B_n \le \theta\}$$

$$= 1 - \lim_{n\to\infty} \inf \Pr\{B_n < -\theta\}$$

$$= 1 - \underline{\nu}(-\theta^+).$$

So $\underline{\nu}(-\theta^+)=1-(\overline{-\nu})(\theta)$. Then

$$\overline{V}_{(1-\gamma)^-} \stackrel{\Delta}{=} \sup\{\theta: \underline{\nu}(\theta) < 1-\gamma\}$$

$$\geq \sup\{\theta: \underline{\nu}(\theta^-) < 1-\gamma\}$$

$$= \sup\{-\hat{\theta}: \underline{\nu}(-\hat{\theta}^+) < 1-\gamma\}$$

$$= \sup\{-\hat{\theta}: 1-(\overline{-\nu})(\hat{\theta}) < 1-\gamma\}$$

$$= -\inf\{\hat{\theta}: (\overline{-\nu})(\hat{\theta}) > \gamma\}$$

$$= -\sup\{\hat{\theta}: (\overline{-\nu})(\hat{\theta}) \leq \gamma\}$$

$$= -(\overline{-V})_\gamma$$

where the inequality follows from $\underline{\nu}(\theta)\geq\underline{\nu}(\theta^-)$. Finally, to show (8), we observe from (6) that $(\overline{U+V})_\delta + (\underline{-V})_\gamma \leq (\overline{U+V-V})_{\delta+\gamma} = \overline{U}_{\delta+\gamma}$. Hence,

$$(\overline{U+V})_\delta \leq \overline{U}_{\delta+\gamma} - (\underline{-V})_\gamma.$$

Using (9), we have the desired result. ■

If we take $\delta=\gamma=0$ in (5) and (7), we obtain

$$(\underline{U+V}) \geq \underline{U}+\underline{V}, \text{ and } (\underline{U+V}) \leq \underline{U}+\overline{V},$$

which mean that the liminf in probability of a sequence of random variables A_n+B_n is upper [resp. lower] bounded by the liminf in probability of A_n plus the limsup [resp. liminf] in probability of B_n. This fact is used in [11] to show that

$$\underline{H}(\boldsymbol{Y}) - \overline{H}(\boldsymbol{Y}|\boldsymbol{X}) \leq \underline{I}(\boldsymbol{X};\boldsymbol{Y}) \leq \underline{H}(\boldsymbol{Y}) - \underline{H}(\boldsymbol{Y}|\boldsymbol{X}),$$

which is a special case of property 3 in Lemma 2.

The next lemmas will show some of the analogous properties of the generalized information measures.

Lemma 2.

For δ, γ, $\delta+\gamma\in[0,1)$, the following statements hold.

1. $\overline{H}_\delta(\boldsymbol{X})\geq 0$. $\overline{H}_\delta(\boldsymbol{X})=0$ if and only if the sequence $\{X^n=(X_1^{(n)},\ldots,X_n^{(n)})\}_{n=1}^{\infty}$ is ultimately deterministic (in probability).

   (This property also applies to $\underline{H}_\delta(\boldsymbol{X})$, $\overline{I}_\delta(\boldsymbol{X};\boldsymbol{Y})$, $\underline{I}_\delta(\boldsymbol{X};\boldsymbol{Y})$, $\overline{D}_\delta(\boldsymbol{X}\|\boldsymbol{Y})$, and $\underline{D}_\delta(\boldsymbol{Y}\|\hat{\boldsymbol{X}})$.)
2. $\underline{I}_\delta(\boldsymbol{X};\boldsymbol{Y})=\underline{I}_\delta(\boldsymbol{Y};\boldsymbol{X})$ and $\overline{I}_\delta(\boldsymbol{X};\boldsymbol{Y})=\overline{I}_\delta(\boldsymbol{Y};\boldsymbol{X})$.
3.

$$\underline{I}_\delta(\boldsymbol{X};\boldsymbol{Y}) \leq \underline{H}_{\delta+\gamma}(\boldsymbol{Y}) - \underline{H}_\gamma(\boldsymbol{Y}|\boldsymbol{X}), \tag{10}$$

$$\underline{I}_\delta(\boldsymbol{X};\boldsymbol{Y}) \leq \overline{H}_{\delta+\gamma}(\boldsymbol{Y}) - \overline{H}_\gamma(\boldsymbol{Y}|\boldsymbol{X}), \tag{11}$$

$$\overline{I}_\gamma(\boldsymbol{X};\boldsymbol{Y}) \leq \overline{H}_{\delta+\gamma}(\boldsymbol{Y}) - \underline{H}_\delta(\boldsymbol{Y}|\boldsymbol{X}), \tag{12}$$

$$\underline{I}_{\delta+\gamma}(\boldsymbol{X};\boldsymbol{Y}) \geq \underline{H}_\delta(\boldsymbol{Y}) - \overline{H}_{(1-\gamma)^-}(\boldsymbol{Y}|\boldsymbol{X}), \tag{13}$$

and

$$\overline{I}_{\delta+\gamma}(\boldsymbol{X};\boldsymbol{Y}) \geq \overline{H}_\delta(\boldsymbol{Y}) - \overline{H}_{(1-\gamma)^-}(\boldsymbol{Y}|\boldsymbol{X}) \tag{14}$$

4. $0\leq\underline{H}_\delta(\boldsymbol{X})\leq\overline{H}_\delta(\boldsymbol{X})\leq\log|\mathcal{X}|$, where each $X_i^{(n)}\in\mathcal{X}$, $i=1,\ldots,n$ and $n=1,2,\ldots$, and $\mathcal{X}$ is finite.
5. $\underline{I}_\delta(\boldsymbol{X},\boldsymbol{Y};\boldsymbol{Z})\geq\underline{I}_\delta(\boldsymbol{X};\boldsymbol{Z})$.

Proof:

Property 1 holds because

$$\Pr\{-\frac{1}{n}\log P_{X^n}(X^n) < 0\} = 0,$$

$$\Pr\{\frac{1}{n}\log\frac{dP_{X^n}}{dP_{\hat{X}^n}}(X^n) < -\delta\} \leq \exp\{-\delta n\}.$$

and

$$\Pr\{\frac{1}{n}\log\frac{dP_{X^nY^n}}{d(P_{X^n}\times P_{Y^n})}(X^n,Y^n) < -\delta\} \leq \exp\{-\delta n\}.$$

Property 2 is an immediate consequence of the definition.

To show the inequalities in property 3 we first remark that

$$\frac{1}{n}h_{Y^n}(Y^n) = \frac{1}{n}i_{(X^n,Y^n)}(X^n,Y^n) + \frac{1}{n}h_{(X^n,Y^n)}(Y^n|X^n),$$

where $(1/n)h_{(X^n,Y^n)}(Y^n|X^n)\stackrel{\Delta}{=}-(1/n)\log P_{Y^n|X^n}(Y^n|X^n)$. With this fact, (10) follows directly from (5), (11) and (12) follow from (6), (13) follows from (7), and (14) follows from (8).

Property 4 follows from the fact that $\overline{H}_\delta(\cdot)$ is non-decreasing in δ: $\overline{H}_\delta(\boldsymbol{X})\leq\overline{H}_1(\boldsymbol{X})=\overline{H}(\boldsymbol{X})$, and that $\overline{H}(\boldsymbol{X})$ is the minimum achievable (i.e., with asymptotically negligible probability of decoding error) fixed-length coding rate for $\boldsymbol{X}$ as seen in [3, Theorem 3.2] and [10].

Property 5 can be proved using the fact that

$$\frac{1}{n}i_{(X^n,Y^n,Z^n)}(X^n,Y^n,Z^n) = \frac{1}{n}i_{(X^n,Z^n)}(X^n,Z^n) + \frac{1}{n}i_{(X^n,Y^n,Z^n)}(Y^n;Z^n|X^n).$$

By applying (5), and letting $\gamma=0$, we obtain the desired result. ■

Lemma 3. (Data processing lemma)

Fix $\delta\in[0,1)$. Suppose that for every n, X_1^n and X_3^n are conditionally independent given X_2^n. Then

$$\underline{I}_\delta(\boldsymbol{X}_1;\boldsymbol{X}_3)\le\underline{I}_\delta(\boldsymbol{X}_1;\boldsymbol{X}_2).$$

Proof:

By property 5, we get

$$\underline{I}_\delta(\boldsymbol{X}_1;\boldsymbol{X}_3)\le\underline{I}_\delta(\boldsymbol{X}_1;\boldsymbol{X}_2,\boldsymbol{X}_3)=\underline{I}_\delta(\boldsymbol{X}_1;\boldsymbol{X}_2),$$

where the equality holds because

$$\frac{1}{n}\log\frac{dP_{X_1^nX_2^nX_3^n}}{d(P_{X_1^n}\times P_{X_2^nX_3^n})}(X_1^n,X_2^n,X_3^n)=\frac{1}{n}\log\frac{dP_{X_1^nX_2^n}}{d(P_{X_1^n}\times P_{X_2^n})}(X_1^n,X_2^n).$$

■

Lemma 4. (Optimality of independent inputs)

Fix $\delta\in[0,1)$. Consider a finite alphabet, discrete memoryless channel – i.e., $P_{Y^n|X^n}=\prod_{i=1}^n P_{Y_i|X_i}$, for all n. For any input $\boldsymbol{X}$ and its corresponding output $\boldsymbol{Y}$,

$$\underline{\boldsymbol{I}}_\delta(\boldsymbol{X},\boldsymbol{Y})\le\underline{\boldsymbol{I}}_\delta(\overline{\boldsymbol{X}};\overline{\boldsymbol{Y}})=\underline{\boldsymbol{I}}(\overline{\boldsymbol{X}};\overline{\boldsymbol{Y}}),$$

where $\overline{\boldsymbol{Y}}$ is the output due to $\overline{\boldsymbol{X}}$, which is an independent process with the same first order statistics as $\boldsymbol{X}$, i.e., $P_{\overline{X}^n}=\prod_{i=1}^n P_{X_i}$.

Proof:

First, we observe that

$$\frac{1}{n}\log\frac{dP_{Y^n|X^n}}{dP_{Y^n}}(X^n,Y^n)+\frac{1}{n}\log\frac{dP_{Y^n}}{dP_{\overline{Y}^n}}(X^n,Y^n)$$

$$=\frac{1}{n}\log\frac{dP_{Y^n|X^n}}{dP_{\overline{Y}^n}}(X^n,Y^n).$$

In other words,

$$\frac{1}{n}\log\frac{dP_{X^nY^n}}{d(P_{X^n}\times P_{Y^n})}(X^n,Y^n)+\frac{1}{n}\log\frac{dP_{Y^n}}{dP_{\overline{Y}^n}}(X^n,Y^n)$$

$$=\frac{1}{n}\log\frac{dP_{\overline{X}^n\overline{Y}^n}}{d(P_{\overline{X}^n}\times P_{\overline{Y}^n})}(X^n,Y^n).$$

By evaluating the above terms under $P_{X^nY^n}$ and letting

$$\overline{z}(\theta)\triangleq\limsup_{n\to\infty}P_{X^nY^n}\{\frac{1}{n}\log\frac{dP_{\overline{X}^n\overline{Y}^n}}{d(P_{\overline{X}^n}\times P_{\overline{Y}^n})}(X^n,Y^n)\le\theta\},$$

and

$$\underline{Z}_\delta(\overline{\boldsymbol{X}},\overline{\boldsymbol{Y}})\triangleq\sup\{\theta:\overline{z}(\theta)\le\delta\},$$

we obtain from (5) (with $\gamma=0$) that

$$\underline{Z}_\delta(\overline{\boldsymbol{X}},\overline{\boldsymbol{Y}})\ge\underline{I}_\delta(\boldsymbol{X},\boldsymbol{Y})+\underline{D}(\boldsymbol{Y}\|\overline{\boldsymbol{Y}})\ge\underline{I}_\delta(\boldsymbol{X};\boldsymbol{Y}),$$

since $\underline{D}(\boldsymbol{Y}\|\overline{\boldsymbol{Y}})\ge0$ by property 1 of Lemma 2.

Note that the summable property of $(1/n)\log[dP_{\overline{X}^n\overline{Y}^n}/d(P_{\overline{X}^n}\times P_{\overline{Y}^n})](X^n,Y^n)$ (i.e., it is equal to $(1/n)\sum_{i=1}^n\log[dP_{\overline{X}_i\overline{Y}_i}/d(P_{\overline{X}_i}\times P_{\overline{Y}_i})](X_i,Y_i)$), the Chebyshev inequality and the finiteness of the channel alphabets imply

$$\underline{I}(\overline{\boldsymbol{X}};\overline{\boldsymbol{Y}})=\underline{I}_\delta(\overline{\boldsymbol{X}};\overline{\boldsymbol{Y}})\text{ and }\underline{Z}(\overline{\boldsymbol{X}};\overline{\boldsymbol{Y}})=\underline{Z}_\delta(\overline{\boldsymbol{X}};\overline{\boldsymbol{Y}}).$$

It finally remains to show that

$$\underline{I}(\overline{\boldsymbol{X}};\overline{\boldsymbol{Y}})\ge\underline{Z}(\overline{\boldsymbol{X}};\overline{\boldsymbol{Y}}),$$

which is proved in [11, Theorem 10].

■

IV. EXAMPLES FOR THE COMPUTATION OF ε-CAPACITY

In [11], Verdú and Han establish the general formulas for channel capacity and ε-capacity. In terms of the ε-inf-information rate, the expression of the ε-capacity becomes

$$\sup_X\underline{I}_{\varepsilon^-}(\overline{\boldsymbol{X}};\overline{\boldsymbol{Y}})\le C_\varepsilon\le\sup_X\underline{I}_\varepsilon(\overline{\boldsymbol{X}};\overline{\boldsymbol{Y}}),$$

where $\varepsilon\in(0,1)$.

We now provide examples for the computation of C_ε. They are basically an extension of some of the examples provided in [11] for the computation of channel capacity. In this section, we assume that all the logarithms are in base 2.

Let the alphabet be binary $\mathcal{X}=\mathcal{Y}=\{0,1\}$, and let every output be given by

$$Y_i=X_i\oplus Z_i$$

where $\oplus$ represents the addition operation modulo-2 and $\boldsymbol{Z}$ is an *arbitrary* binary random process independent of $\boldsymbol{X}$.

To compute the ε-capacity we use the results of property 3 in Lemma 2:

$$\underline{I}_{\varepsilon^-}(\overline{\boldsymbol{X}};\overline{\boldsymbol{Y}})\ge\underline{H}_0(\boldsymbol{Y})-\overline{H}_{(1-\varepsilon^-)^-}(\boldsymbol{Y}|\boldsymbol{X})=\underline{H}_0(\boldsymbol{Y})-\overline{H}_{(1-\varepsilon)}(\boldsymbol{Y}|\boldsymbol{X}),\tag{15}$$

and

$$\underline{I}_\varepsilon(\underline{\boldsymbol{X}};\underline{\boldsymbol{Y}})\le\min\{\underline{H}_{\varepsilon+\gamma}(\boldsymbol{Y})-\underline{H}_\gamma(\boldsymbol{Y}|\boldsymbol{X}),\overline{H}_{\varepsilon+\gamma}(\boldsymbol{Y})-\overline{H}_\gamma(\boldsymbol{Y}|\boldsymbol{X})\},\tag{16}$$

where $\varepsilon \geq 0$, $\gamma \geq 0$ and $1 > \varepsilon + \gamma$. The lower bound in (15) follows directly from (13) (by taking $\delta = 0$ and $\gamma = \varepsilon^-$). The upper bounds in (16) follow from (10) and (11) respectively.

$$C_\varepsilon \leq \sup_X I_\varepsilon(X;Y)$$

$$\leq \sup_X \{\overline{H}_{\varepsilon+\gamma}(Y) - \overline{H}_\gamma(Y|X)\}.$$

Since the above inequality holds for all $0 \leq \gamma < 1-\varepsilon$, we have:

$$C_\varepsilon \leq \inf_{0 \leq \gamma < 1-\varepsilon} \sup_X \{\overline{H}_{\varepsilon+\gamma}(Y) - \overline{H}_\gamma(Y|X)\}$$

$$\leq \inf_{0 \leq \gamma < 1-\varepsilon} \{\sup_X \overline{H}_{\varepsilon+\gamma}(Y) - \inf_X \overline{H}_\gamma(Y|X)\}.$$

By the symmetry of the channel, $\overline{H}_\gamma(Y|X) = \overline{H}_\gamma(Z)$ which is independent of X. Hence,

$$C_\varepsilon \leq \inf_{0 \leq \gamma < 1-\varepsilon} \{\sup_X \overline{H}_{\varepsilon+\gamma}(Y) - \overline{H}_\gamma(Z)\}$$

$$\leq \inf_{0 \leq \gamma < 1-\varepsilon} \{\log_2 2 - \overline{H}_\gamma(Z)\} = \inf_{0 \leq \gamma < 1-\varepsilon} \{1 - \overline{H}_\gamma(Z)\},$$

where the last step follows by taking a Bernoulli uniform input. Since $1-\overline{H}_\gamma(Z)$ is non-increasing in γ,

$$C_\varepsilon \leq 1 - \overline{H}_{(1-\varepsilon)^-}(Z).$$

(Note that the superscript "–" indicates a strict inequality in the definition of $\overline{H}_\gamma(\cdot)$; this is consistent with the condition $\gamma + \varepsilon < 1$.)

On the other hand, we can derive the lower bound to C_ε by choosing a Bernoulli uniform input in (15). We thus obtain

$$1 - \overline{H}_{(1-\varepsilon)}(Z) \leq C_\varepsilon \leq 1 - \overline{H}_{(1-\varepsilon)^-}(Z).$$

Note that there are actually two upper bounds (16). In this example, the first upper bound $1-\underline{H}_{(1-\varepsilon)^-}(Z)$ (which is no less than $1-\overline{H}_{(1-\varepsilon)^-}(Z)$) is a looser upper bound, and hence, can be omitted. In addition, we demonstrate in the above derivation that the computation of the upper bound to C_ε involves in general the infimum operation over the parameter γ. Therefore, if the optimizing input distribution does not have a "nice" property (such as independence and uniformity), then the computation of (17) may be complicated in general.

Remark:

An *alternative* method to compute C_ε is to derive the channel sup-spectrum in terms of the inf-spectrum of the noise process. Under the optimizing equally likely Bernoulli input X^* we can write

$$\overline{i}_{(X^*;Y)}(\theta) = \limsup_{n\to\infty} \Pr\{\frac{1}{n}\log \frac{P_{Y^n|X^n}(Y^n|X^n)}{P_{Y^n}(Y^n)} \leq \theta\}$$

$$= \limsup_{n\to\infty} \Pr\{\frac{1}{n}\log P_{Z^n}(Z^n) - \frac{1}{n}\log P_{Y^n}(Y^n) \leq \theta\}$$

$$= \limsup_{n\to\infty} \Pr\{\frac{1}{n}\log P_{Z^n}(Z^n) \leq \theta - 1\}$$

$$= \limsup_{n\to\infty} \Pr\{-\frac{1}{n}\log P_{Z^n}(Z^n) \geq 1 - \theta\}$$

$$= 1 - \underline{h}_Z((1-\theta)^-).$$

Hence,

$$\underline{I}_\varepsilon(X^*;Y) = \sup\{\theta: 1 - \underline{h}_Z((1-\theta)^-) \leq \varepsilon\}$$

$$= \sup\{\theta: \underline{h}_Z((1-\theta)^-) \geq 1-\varepsilon\}$$

$$= \sup\{(1-\beta): \underline{h}_Z(\beta^-) \geq 1-\varepsilon\}$$

$$= 1 + \sup\{(-\beta): \underline{h}_Z(\beta^-) \geq 1-\varepsilon\}$$

$$= 1 - \inf\{\beta: \underline{h}_Z(\beta^-) \geq 1-\varepsilon\}$$

$$= 1 - \sup\{\beta: \underline{h}_Z(\beta^-) < 1-\varepsilon\}$$

$$= 1 - \overline{H}_{(1-\varepsilon)^-}(Z).$$

Similarly,

$$\underline{I}_{\varepsilon^-}(X^*;Y) = 1 - \overline{H}_{(1-\varepsilon)}(Z).$$

Therefore,

$$1 - \overline{H}_{(1-\varepsilon)}(Z) = \underline{I}_{\varepsilon^-}(X^*;Y) \leq C_\varepsilon \leq \underline{I}_\varepsilon(X^*;Y) = 1 - \overline{H}_{(1-\varepsilon)^-}(Z).$$

Example 1.

Let Z be an all-zero sequence with probability β and Bernoulli (with parameter p) with probability $1-\beta$. Then the sequence of random variables $(1/n)h_{Z^n}(Z^n)$ converges to atoms 0 and $h_b(p) \triangleq -p\log p - (1-p)\log(1-p)$ with respective masses β and $1-\beta$. The resulting $\underline{h}_Z(\theta)$ is depicted in Fig. 2. From (18), we obtain $\overline{i}_{(X,Y)}(\theta)$ as shown in Fig. 3.

Therefore,

$$C_\varepsilon = \begin{cases} 1 - h_b(p), & \text{if } 0 < \varepsilon < 1-\beta; \\ 1, & \text{if } 1-\beta < \varepsilon < 1. \end{cases}$$

When $\varepsilon = 1-\beta$, C_ε lies somewhere between $1-h_b(p)$ and 1.

Example 2.

If Z is a non-stationary binary independent

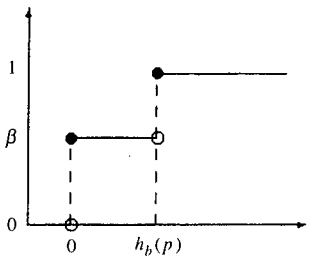


Fig. 2. The spectrum of $(1/n)h_{Z^n}(Z^n)$ for Example 1.

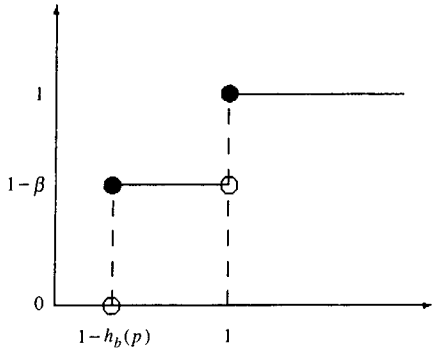


Fig. 3. The spectrum of $(1/n)i_{(X^n, Y^n)}(X^n; Y^n)$ for Example 1.

sequence with $\Pr\{Z_1=1\}=p_i$, then by the uniform boundedness (in i) of the variance of random variable $-\log P_{Z_i}(Z_i)$, namely,

$$\begin{aligned}\mathrm{Var}[-\log P_{Z_i}(Z_i)] &\le E[(\log P_{Z_i}(Z_i))^2] \\ &\le \sup_{0<p_i<1} p_i(\log p_i)^2+(1-p_i)(\log(1-p_i))^2 \\ &\le 1,\end{aligned}$$

we have (by Chebyshev's inequality)

$$\Pr\left\{\left|-\frac{1}{n}\log P_{Z^n}(Z^n)-\frac{1}{n}\sum_{i=1}^{n} H(Z_i)\right|<\gamma\right\}\to 0,$$

for any $\gamma>0$. Therefore, $\overline{H}_{(1-\varepsilon)}(\mathbf{Z})$ is independent of ε, and C_ε is equal to 1 minus the largest cluster point of $(1/n)\sum_{i=1}^{n} H(Z_i)$, i.e.,

$$\overline{H}_{(1-\varepsilon)}(\mathbf{Z})=\limsup_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n} H(Z_i),$$

and

$$C_\varepsilon=1-\overline{H}(\mathbf{Z})=1-\limsup_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n} H(Z_i),$$

where $H(Z_i)=h_b(p_i)$. This result is illustrated in Figs. 4 and 5.

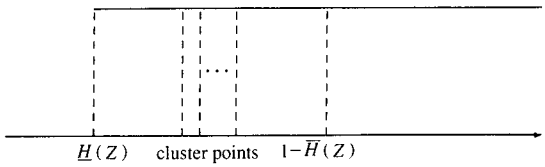


Fig. 4. The spectrum of $(1/n)h_{Z^n}(Z^n)$ for Example 2.

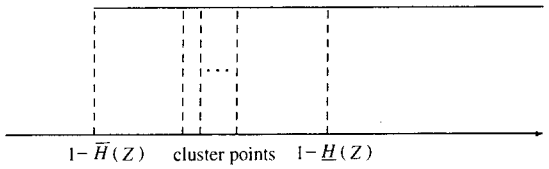


Fig. 5. The spectrum of $(1/n)i_{(X^n, Y^n)}(X^n; Y^n)$ for Example 2.

V. CONCLUSIONS

In light of the work of Han and Verdú in [10] and [11], generalized entropy, mutual-information, and divergence rates are proposed. The properties of each of these information quantities are analyzed, and examples illustrating the computation of the ε-capacity of channels with arbitrary additive noise are presented.

In [3], we use these information measures to prove a generalized version of the Asymptotic Equipartition Property (AEP) and general source coding and hypothesis testing theorems.

ACKNOWLEDGMENT

The authors would like to thank Prof. S. Verd\ú for his valuable advice and constructive criticism which helped improve the paper.

REFERENCES

1. Billingsley, P. *Probability and Measure*, Wiley, New York (1986).
2. Blahut, R.E. *Principles and Practice of Information Theory*, Addison Wesley, Massachusetts (1988).
3. Chen P.-N. and F. Alajaji, "Generalized Source Coding Theorems and Hypothesis Testing: Part II – Operational limits," *Journal of the Chinese Institute of Engineers*, Vol. 21, No. 3, May (1998).
4. Chen P.-N. and F. Alajaji, "Strong Converse, Feedback Channel Capacity and Hypothesis Testing," *Journal of the Chinese Institute of Engineers*, Vol. 18, pp. 777-785, November 1995; also in Proceedings of CISS, John Hopkins Univ., MD, USA, March (1995).

5. Chen P.-N. and F. Alajaji, "The Reliability Function of Arbitrary Channels with and Without Feedback," *Proceedings of the 18'th Biennial Symposium on Communications*, Queen's University, Kingston, Ontario, June (1996).
6. Chen, P.-N. "General Formulas for the Neyman-Pearson type-II Error Exponent Subject to Fixed and Exponential type-I Error Bounds," *IEEE Transactions on Information Theory*, T-42(1): 316-323, January (1996).
7. Cover T.M. and J.A. Thomas, *Elements of Information Theory*, Wiley, New York (1991).
8. Csiszár I. and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems, Academic*, New York, 1981.
9. Gray, R.M. *Entropy and Information Theory*. Springer-Verlag, New York (1990).
10. Han T.S. and S. Verdú, "Approximation Theory of Output Dtatistics," *IEEE Transactions on Information Theory*, IT-39(3): 752-772, May (1993).
11. Verdú S. and T.S. Han, "A General Formula for Channel Capacity," *IEEE Transactions on Information Theory*, IT-40(4) : 1147-1157, Jul. (1994).

Discussions of this paper may appear in the discussion section of a future issue. All discussions should be submitted to the Editor-in-Chief.

Manuscript Received: June 25, 1997
Revision Received: Jan. 11,1998
and Accepted: Jan. 24, 1998

來源編碼定理與檢定測試的一般定理：第一部份 —訊息量度

陳伯寧

國立交通大學電信工程系所

Fady Alajaji
Queen's University
Kingston, ON K7L 3N6, Canada

摘 要

本論文將介紹ε-熵率、ε-相互訊息率與ε-偏差率的表示式，這三個表示式是擴展Han與Verdú的極下／極上－熵／相互訊息／偏差率而得到的，式中包含了對應近似訊息機率譜。我們將分析它們的代數性質，舉例說明其於ε-管道傳輸能力的計算上之應用。在此研究的第二部份，這三個公式將被用來證明區塊碼來源編碼一般定理與紐門皮爾森檢定測試第二類錯誤指數之一般公式。

關鍵詞：消息理論、近似平均分割性質、來源編碼定理、檢定測試、紐門皮爾森錯誤指數。

Journal of the Chinese Institute of Engineers, Vol. 21, No. 3, pp. 293-303 (1998)

GENERALIZED SOURCE CODING THEOREMS AND HYPOTHESIS TESTING: PART II -- OPERATIONAL LIMITS

Po-Ning Chen*
Dept. of Communications Engineering
National Chiao Tung University
Hsin Chu, Taiwan 300, R.O.C.

Fady Alajaji
Dept. of Mathematics and Statistics
Queen's University
Kingston, Ontario K7L 3N6, Canada

Key Words: Shannon theory, AEP, source coding theorems, hypothesis testing, Neyman-Pearson error exponent.

ABSTRACT

In light of the information measures introduced in Part I, a generalized version of the Asymptotic Equipartition Property (AEP) is proved. General fixed-length data compaction and data compression (source coding) theorems for arbitrary finite-alphabet sources are also established. Finally, the general expression of the Neyman-Pearson type-II error exponent subject to upper bounds on the type-I error probability is examined.

I. INTRODUCTION

In Part I of this paper [3], generalized versions of the inf/sup-entropy/information/divergence rates of Han and Verdú were proposed and analyzed. Equipped with these information measures, we herein demonstrate a generalized Asymptotic Equipartition Property (AEP) Theorem and establish expressions for the infimum $(1-\varepsilon)$-achievable (fixed-length) coding rate of an arbitrary finite-alphabet source $\boldsymbol{X}$. These expressions turn out to be the *counterparts* of the ε-capacity formulas in [11, Theorem 6]. We also prove a general data compression theorem; this theorem extends a recent rate-distortion theorem [9, Theorem 10(a)] by Steinberg and Verdú (cf the remarks at the end of Sections II.1 and II.2).

The Neyman-Pearson hypothesis testing problem examined in [4] is revisited in light of the generalized divergence measures.

Since this work is a continuation of [3], we refer the reader to [3] for the technical definitions of the information measures used in this paper.

II. GENERAL SOURCE CODING THEOREMS

The role of a source code is to represent the output of a source efficiently. This is achieved by introducing some controlled distortion into the source, hence reducing its intrinsic information content. There are two classes of source codes: data compaction codes and data compression codes [2]. The objective of both types of codes is to minimize the source description rate of the codes subject to a fidelity criterion constraint. In the case of data compaction, the fidelity criterion consists of the probability of decoding error Pe. If Pe is made arbitrarily small, we obtain a traditional error-free (or lossless) source coding system. Data compression codes are a larger class of codes in the sense that the fidelity

*Correspondence addressee

criterion used in the coding scheme is a general distortion measure. We herein demonstrate data compaction and data compression theorems for arbitrary (not necessarily stationary ergodic, information stable, etc.) sources.

In this section, we assume that the source alphabet $\mathcal{X}$ is *finite*[1].

1. Data compaction coding theorem

Definition 1. (e.g. [2])

A block code for data compaction is a set C_n consisting of $M \triangleq |C_n|$ codewords of blocklength n:

$$C_n \triangleq \{c_1^n, c_2^n, ..., c_M^n\},$$

where each n-tuple $c_i^n \in \mathcal{X}^n$, $i=1, 2, ..., M$.

Definition 2.

Fix $1 \geq \varepsilon \geq 0$. R is a $(1-\varepsilon)$-achievable data compaction rate for a source $\boldsymbol{X}$ if there exists a sequence of data compaction codes C_n with

$$\limsup_{n\to\infty} \frac{1}{n} \log |C_n| = R,$$

and

$$\limsup_{n\to\infty} Pe(C_n) \leq 1-\varepsilon,$$

where $Pe(C_n) \triangleq \Pr(X^n \notin C_n)$ is the probability of decoding error. The infimum $(1-\varepsilon)$-achievable data compaction rate for $\boldsymbol{X}$ is denoted by $T_{1-\varepsilon}(\boldsymbol{X})$.

For discrete memoryless sources, the data compaction theorem is proved by choosing the codebook C_n to be the (*weakly*) *typical set* [2]and applying the Asymptotic Equipartition Property (AEP) [2] [5] which states that $(1/n)h_{X^n}(X^n)$ converges to $H(X)$ with probability one (and hence in probability). The AEP -- which implies that the probability of the typical set is close to one for sufficiently large n -- also holds for stationary ergodic sources [5]. It is however invalid for more general sources -- e.g., nonstationary, nonergodic sources. We herein demonstrate a generalized AEP theorem.

Theorem 1. (Generalized AEP)

Fix $1 > \varepsilon > 0$. Given an arbitrary source $\boldsymbol{X}$, define

$$\mathcal{T}_n[R] \triangleq \{x^n \in \mathcal{X}^n : -\frac{1}{n} \log P_{X^n}(x^n) \leq R\}.$$

Then $(\forall \gamma > 0)$ such that the following statements hold.

1.
$$\liminf_{n\to\infty} \Pr\{\mathcal{T}_n[\overline{H}_\varepsilon(\boldsymbol{X}) - \gamma]\} \leq \varepsilon \quad (1)$$

2.
$$\liminf_{n\to\infty} \Pr\{\mathcal{T}_n[\overline{H}_\varepsilon(\boldsymbol{X}) + \gamma]\} > \varepsilon \quad (2)$$

3. The number of elements in $\mathcal{T}_n[\overline{H}_\varepsilon(\boldsymbol{X})]$, denoted by $|\mathcal{T}_n[\overline{H}_\varepsilon(\boldsymbol{X})]|$, satisfies

$$|\mathcal{T}_n[\overline{H}_\varepsilon(\boldsymbol{X})+\gamma] - \mathcal{T}_n[\overline{H}_\varepsilon(\boldsymbol{X})-\gamma]| \leq \exp\{n(\overline{H}_\varepsilon(\boldsymbol{X})+\gamma)\}. \quad (3)$$

4. $(\forall \gamma > 0)(\exists \rho = \rho(\gamma) > 0$, N_0 and a subsequence $\{n_j\}_{j=1}^{\infty}$ such that $\forall n_j > N_0)$,

$$|\mathcal{T}_{n_j}[\overline{H}_\varepsilon(\boldsymbol{X})+\gamma] - \mathcal{T}_{n_j}[\overline{H}_\varepsilon(\boldsymbol{X})-\gamma]| > \rho(\gamma)\exp\{n_j(\overline{H}_\varepsilon(\boldsymbol{X})-\gamma)\}, \quad (4)$$

where the operation $A-B$ between two sets A and B is defined by $A-B \triangleq A \cap B^c$, with B^c denoting the complement set of B.

Proof:

(1) and (2) follow from the definitions. For (3), we have

$$1 \geq \sum_{x^n \in \mathcal{T}_n[\overline{H}_\varepsilon(\boldsymbol{X})+\gamma] - \mathcal{T}_n[\overline{H}_\varepsilon(\boldsymbol{X})-\gamma]} P_{X^n}(x^n)$$

$$\geq \sum_{x^n \in \mathcal{T}_n[\overline{H}_\varepsilon(\boldsymbol{X})+\gamma] - \mathcal{T}_n[\overline{H}_\varepsilon(\boldsymbol{X})-\gamma]} \exp\{-n(\overline{H}_\varepsilon(\boldsymbol{X})+\gamma)\}$$

$$\geq \left|\mathcal{T}_n[\overline{H}_\varepsilon(\boldsymbol{X})+\gamma] - \mathcal{T}_n[\overline{H}_\varepsilon(\boldsymbol{X})-\gamma]\right| \exp\{-n(\overline{H}_\varepsilon(\boldsymbol{X})+\gamma)\}.$$

It remains to show (4). (2) implies that there exist $\rho = \rho(\gamma) > 0$ and N_1 such that for all $n > N_1$,

$$\Pr\{\mathcal{T}_n[\overline{H}_\varepsilon(\boldsymbol{X})+\gamma]\} > \varepsilon + 2\rho(\gamma).$$

Furthermore, (1) implies that for the previously chosen $\rho(\gamma)$, there exist N_2 and a subsequence $\{n_j\}_{j=1}^{\infty}$ such that for all $n_j > N_2$,

$$\Pr\{\mathcal{T}_{n_j}[\overline{H}_\varepsilon(\boldsymbol{X})-\gamma]\} < \varepsilon + \rho(\gamma).$$

Therefore, for all $n_j > N_0 \triangleq \max(N_1, N_2)$,

$$\rho(\gamma) < \Pr\{\mathcal{T}_{n_j}[\overline{H}_\varepsilon(\boldsymbol{X})+\gamma] - \mathcal{T}_{n_j}[\overline{H}_\varepsilon(\boldsymbol{X})-\gamma]\}$$

$$< \left|\mathcal{T}_{n_j}[\overline{H}_\varepsilon(\boldsymbol{X})+\gamma] - \mathcal{T}_{n_j}[\overline{H}_\varepsilon(\boldsymbol{X})-\gamma]\right| \exp\{-n_j(\overline{H}_\varepsilon(\boldsymbol{X})-\gamma)\}.$$

Comment:

With the illustration depicted in Fig. 1, we

[1] Actually, the theorems in this section also apply for sources with countable alphabets. We assume finite alphabets in order to avoid uninteresting cases (such as $\overline{H}_\varepsilon(\boldsymbol{X}) = \infty$) that might arise with countable alphabets.

can clearly deduce that Theorem 1 is *indeed* a generalized version of the AEP since:

- The set

$$B-A \stackrel{\Delta}{=} \mathcal{T}_n[\overline{H}_\varepsilon(X)+\gamma]-\mathcal{T}_n[\overline{H}_\varepsilon(X)-\gamma]$$

$$= \{x^n \in \mathcal{X}^n: \left| -\frac{1}{n}\log P_{X^n}(x^n) - \overline{H}_\varepsilon(X) \right| \le \gamma\}$$

  is nothing but the typical set.

- (1) and (2) $\Rightarrow$ that $q \stackrel{\Delta}{=} \Pr(B-A)>0$ infinitely often.
- (3) and (4) $\Rightarrow$ that the number of sequences in $B-A$ (the dashed region) is approximatively equal to $\exp\{n\overline{H}_\varepsilon(X)\}$, and the probability of each sequence in $B-A$ is $\approx q \times \exp\{-n\overline{H}_\varepsilon(X)\}$.
- In particular, if X is a stationary ergodic source, then $\overline{H}_\varepsilon(X)$ is independent of ε and $\overline{H}_\varepsilon(X)=\underline{H}_\varepsilon(X)=H\ \forall \varepsilon(0,1)$, where H is the source entropy rate

$$H = \lim_{n\to\infty} \frac{1}{n} E_{P_{X^n}}[-\log P_{X^n}(X^n)].$$

  In this case, (1)-(2) and the fact that $\overline{H}_\varepsilon(X)=\underline{H}_\varepsilon(X)\ \forall \varepsilon$ imply that the probability q of the typical set $B-A$ is close to one (for n sufficiently large), and (3) and (4) imply that there are about e^{nH} typical sequences of length n, each with probability about e^{-nH}. Hence we obtain the conventional AEP (cf [3, Theorem 3.1.2] or [2, Theorem 3.4.2]).

We now apply Theorem 1 to prove a *general* data compaction theorem for block codes.

Theorem 2. (General data compaction theorem)
Fix $1>\varepsilon>0$. For any source X,

$$\overline{H}_{\varepsilon^-}(X) \le T_{1-\varepsilon}(X) \le \overline{H}_\varepsilon(X).$$

Note that actually $T_{1-\varepsilon}(X)=\overline{H}_{\varepsilon^-}(X)$, since $T_{1-\varepsilon}(X)$ is left-continuous in ε (cf Appendix B).

Proof: *Forward part (achievability)*:
We need to prove the existence of a sequence of block codes C_n with

$$\limsup_{n\to\infty} \frac{1}{n}\log|C_n| < \overline{H}_\varepsilon(X)+2\gamma,$$

and

$$\limsup_{n\to\infty} Pe(C_n) \le 1-\varepsilon.$$

Choose the code to be $C_n=\mathcal{T}_n[\overline{H}_\varepsilon(X)+\gamma]$. Then by definition of $\mathcal{T}_n[\cdot]$,

$$|C_n| = \left|\mathcal{T}_n[\overline{H}_\varepsilon(X)+\gamma]\right| \le \exp\{n(\overline{H}_\varepsilon(X)+\gamma)\}.$$

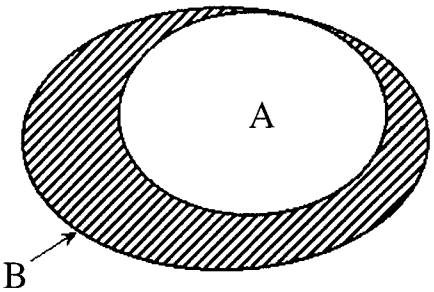


Fig. 1. Illustration of the Generalized AEP Theorem. $A=\mathcal{T}_n[\overline{H}_\varepsilon(X)-\gamma]$, $B=\mathcal{T}_n[\overline{H}_\varepsilon(X)+\gamma]$, and $(B-A)$ is the dashed region.

Therefore

$$\limsup_{n\to\infty} \frac{1}{n}\log|C_n| \le \overline{H}_\varepsilon(X)+\gamma < \overline{H}_\varepsilon(X)+2\gamma.$$

On the other hand,

$$1-Pe(C_n) = \Pr\{C_n\} = \Pr\{\mathcal{T}_n[\overline{H}_\varepsilon(X)+\gamma]\},$$

which implies from (2) that

$$\liminf_{n\to\infty}[1-Pe(C_n)] = \liminf_{n\to\infty} \Pr\{\mathcal{T}_n[\overline{H}_\varepsilon(X)+\gamma]\} > \varepsilon.$$

Hence,

$$\liminf_{n\to\infty} Pe(C_n) < 1-\varepsilon.$$

Accordingly, $T_{1-\varepsilon}(X) < \overline{H}_\varepsilon(X)+2\gamma$ for any $\gamma>0$. This proves the forward part.

To show the converse part, we need the following remark.

Remark:
For all $x^n \in C_n^*$ and $\hat{x}^n \notin C_n^*$,

$$P_{X^n}(x^n) \ge P_{X^n}(\hat{x}^n),$$

where C_n^* is the optimal block code defined as follows: for any block code C_n with $|C_n|=|C_n^*|$, $Pe(C_n^*) \le Pe(C_n)$.

This result follows directly from the definition of C_n^* and the fact that $Pe(C_n^*)=P_{X^n}([C_n^*]^c)$. The above remark indeed points out that the optimal code must be of the shape

$$\{x^n \in \mathcal{X}^n: -\frac{1}{n}\log P_{X^n}(x^n) < R\}$$

$$\subset C_n^* \subset \{x^n \in \mathcal{X}^n: -\frac{1}{n}\log P_{X^n}(x^n) \le R\} \qquad (5)$$

2. *Converse part:* We show that for all codes with code rate

$$R \triangleq \limsup_{n\to\infty} (1/n)\log\left|C_n\right| < \overline{H}_{\varepsilon'}(\boldsymbol{X}),$$

$$\limsup_{n\to\infty} Pe(C_n) > 1-\varepsilon.$$

By definition of $\overline{H}_{\varepsilon'}(\boldsymbol{X})$, there exists $0<\varepsilon'<\varepsilon$ such that $R<\overline{H}_{\varepsilon'}(\boldsymbol{X})\le \overline{H}_{\varepsilon}(\boldsymbol{X})$.

Since $Pe(C_n^*)\le Pe(C_n)$ for C_n^* with the same size as Cn, we only need to show

$$\limsup_{n\to\infty} Pe(C_n^*) > 1-\varepsilon.$$

(5) already gives us the shape of the optimal block code. We claim that the set $\mathcal{T}_n[\overline{H}_{\varepsilon'}(\boldsymbol{X})+\gamma]-\mathcal{T}_n[\overline{H}_{\varepsilon'}(\boldsymbol{X})]$ is not contained in C_n^* for any $\gamma>0$ infinitely often because if it were, then by slightly modifying the proof of (4), it can be shown that there exists $\gamma>0$ such that

$$\left|C_{n_j}^*\right| > \left|\mathcal{T}_{n_j}[\overline{H}_{\varepsilon'}(\boldsymbol{X})+\gamma]-\mathcal{T}_{n_j}[\overline{H}_{\varepsilon'}(\boldsymbol{X})]\right|$$

$$> \rho(\gamma)\exp\{n_j(\overline{H}_{\varepsilon'}(\boldsymbol{X}))\}$$

for some positive $\rho(\gamma)$, subsequence $\{n_j\}_{j=1}^{\infty}$ and sufficiently large j, implying that

$$R \ge \overline{H}_{\varepsilon'}(\boldsymbol{X})$$

This violates the code rate constraint $R<\overline{H}_{\varepsilon'}(\boldsymbol{X})$. Hence, C_n^* is a subset of $\mathcal{T}_n[\overline{H}_{\varepsilon'}(\boldsymbol{X})]$ for all but finitely many n. Consequently,

$$\liminf_{n\to\infty}[1-Pe(C_n^*)] = \liminf_{n\to\infty}\Pr(C_n^*)$$

$$\le \liminf_{n\to\infty}\Pr\{\mathcal{T}_n[\overline{H}_{\varepsilon'}(\boldsymbol{X})]\} \le \varepsilon' < \varepsilon,$$

where the last inequality follows from the definition of $\overline{H}_{\varepsilon'}(\boldsymbol{X})$. This immediately implies that

$$\limsup_{n\to\infty} Pe(C_n^*) > 1-\varepsilon.$$

This proves the converse part. ■

Observations:

- For the sake of clarity, we only considered in Theorem 2 the case where $\varepsilon\in(0,1)$. We can however easily extend the result to the cases where $\varepsilon=0$ and $\varepsilon=1$. By definition, $\overline{H}_0(\boldsymbol{X})=-\infty$ and $\overline{H}_1(\boldsymbol{X})=\infty$. Therefore, to show that Theorem 2 holds for $\varepsilon=0$ and $\varepsilon=1$, it suffices to prove that

$$T_1(\boldsymbol{X}) \le \overline{H}_0(\boldsymbol{X}) \qquad (7)$$

and

$$T_0(\boldsymbol{X}) \ge \overline{H}_1(\boldsymbol{X}) \qquad (8)$$

The validity of (7) follows from the proof of the forward-part of Theorem 2; similarly, (8) can be verified using the same arguments in the proof of the converse-part of Theorem 2.

- Theorem 2 is indeed the *counterpart* of the result on the channel ε-capacity in [11, Theorem 6]. It describes, in terms of the parameter ε, the relationship between the code rate and the ultimate probability of decoding error:

$$Pe \approx 1-\varepsilon \text{ and } R = \overline{H}_{\varepsilon'}(\boldsymbol{X}).$$

- Note that as $\varepsilon\uparrow 1$, $\overline{H}_{\varepsilon'}(\boldsymbol{X})\to\overline{H}_1(\boldsymbol{X})=\overline{H}(\boldsymbol{X})$. Hence, this theorem generalizes the block source coding theorem in [8], which states that the minimum achievable fixed-length source coding rate of any finite-alphabet source is $\overline{H}(\boldsymbol{X})$.
- Consider the special case where $-(1/n)\log P_{X^n}(X^n)$ converges in probability to a constant H; this reduces Theorem 1 to the conventional AEP [3]. In this case, both $\underline{h}_{\boldsymbol{X}}(\cdot)$ and $\overline{h}_{\boldsymbol{X}}(\cdot)$ degenerate to a unit step function:

$$u(\theta-H) = \begin{cases} 1, & \text{if } \theta > \mathrm{H}; \\ 0, & \text{if } \theta < \mathrm{H}, \end{cases}$$

yielding $\underline{H}(\boldsymbol{X})=\overline{H}_{\varepsilon'}(\boldsymbol{X})=\overline{H}(\boldsymbol{X})=H$ for all $\varepsilon\in(0,1)$, where H is the source entropy rate. Hence, our result reduces to the conventional source coding theorem for information stable sources [10, Theorem 1].

- More generally, if $-(1/n)\log P_{X^n}(X^n)$ converges in probability to a random variable Z whose cumulative distribution function (cdf) is $F_Z(\cdot)$, we have

$$Pe \approx 1-F_Z(R) \text{ for } R=\overline{H}_{\varepsilon'}(\boldsymbol{X})=\underline{H}_{\varepsilon'}(\boldsymbol{X}).$$

Therefore, the relationship between the code rate and the ultimate optimal error probability is also clearly defined.

Example: Consider a binary exchangeable (hence stationary but nonergodic in general [1]) source $\boldsymbol{X}$. Then there exists a distribution G concentrated on the interval $(0,1)$ such that the process X is a mixture of Bernoulli (θ) processes where the parameter $\theta\in\Theta=(0\ 1)$ and has distribution G [1, Corollary 1]. In this case, it can be shown via the ergodic decomposition theorem that $-(1/n)\log P_{X^n}(X^n)$ converges in probability to $Z=h_b(\theta)$ [1] [7], where $h_b(x)\triangleq -x\log_2(x)-(1-x)\log_2(1-x)$ is the binary entropy function. We therefore find that the cdf of Z is $F_Z(z)=P(h_b(\Theta)\le z)$ where θ has distribution G. Finally, note that as $\varepsilon\uparrow 1$, $Pe\to 0$ and

$$\lim_{\varepsilon\uparrow 1}\overline{H}_{\varepsilon'}(\boldsymbol{X}) = \inf[r: dG(h_b(\theta)\le r)=1] \triangleq ess_{\Theta}\sup h_b(\theta).$$

The above equation is indeed the minimum achievable (i.e., with $Pe \to 0$) fixed-length source coding rate for stationary nonergodic sources [6].

Remark:

In this work, the definition that we adopt for the $(1-\varepsilon)$-achievable data compaction rate, is slightly different from the one used in [8, Definition 8]. As a result, our $T_{1-\varepsilon}(\boldsymbol{X})$ is right-continuous with respect to $(1-\varepsilon)$, and is equal to $\overline{H}_{\varepsilon}(\boldsymbol{X})$ for $\varepsilon \in (0,1]$ and 0 for $\varepsilon=0$ (cf Appendix B). In fact, the definition in [8] also yields the same result, which was separately proved by Steinberg and Verdú as a direct consequence of Theorem 10(a) [9] (cf Corollary 3 in [9]). To be precise, their $T_{1-\varepsilon}(\boldsymbol{X})$, denoted by $T_e(1-\varepsilon,\boldsymbol{X})$ in [9], is shown for $0<\varepsilon<1$ to be equal to

$T_e(1-\varepsilon,\boldsymbol{X})=R_v(2(1-\varepsilon))$, (cf Definition 17 in [9])

$$= \inf\{\theta: \limsup_{n\to\infty} P_{X^n}[-\frac{1}{n}\log P_{X^n}(X^n) > \theta] \le 1-\varepsilon\}$$

$$= \inf\{\theta: \liminf_{n\to\infty} P_{X^n}[-\frac{1}{n}\log P_{X^n}(X^n) \le \theta] \ge \varepsilon\}$$

$$= \inf\{\theta: \underline{h}(\theta) \ge \varepsilon\}$$

$$= \sup\{\theta: \underline{h}(\theta) < \varepsilon\}$$

$$= \overline{H}_{\varepsilon}(\boldsymbol{X}).$$

Note that Theorem 10(a) in [9] is a data compression theorem for arbitrary sources which the authors show as a by-product of their results on finite-precision resolvability theory[9]. Here, we establish Theorem 2 in a different and more direct way; it is proven using the generalized entropy measure introduced in [5] and the Generalized AEP (Theorem 1). In the next section, we generalize Theorem 10(a) of [9].

2. Data compression coding theorem

Definition 3. (e.g. [2])

Given a source alphabet $\mathcal{X}$ and a reproduction alphabet $\mathcal{Y}$, a block code for data compression of blocklength n and size M is a mapping $f_n(\cdot)$: $x^n \to y^n$ that results in $||f_n||=M$ codewords of lengt n, where each codeword is a sequence of n reproduction letters.

Definition 4.

A distortion measure $\rho_n(\cdot,\cdot)$ is a mapping

$$\rho_n: x^n \times y^n \to \mathcal{R}^+ \stackrel{\Delta}{=} [0,\infty).$$

We can view the distortion measure as the cost of representing a source n-tuple X^n by a reproduction n-tuple $f_n(X^n)$.

In Theorem 10.(a) of [9], Steinberg and Verdú provide a data compression theorem for arbitrary sources under the restriction that the probability of excessive distortion due to the achievable data compression codes is equal to zero (cf Definitions 30 and 31 in [9]). We herein provide a generalization of their result by relaxing the restriction on the probability of excessive distortion.

Definition 5. (Distortion inf-spectrum and ε-sup-distortion rate)

Let X and $\{\rho_n(\cdot,\cdot)\}_{n\ge 1}$ be given. Let $\boldsymbol{f}(\boldsymbol{X}) \stackrel{\Delta}{=} \{f_n(X^n)\}_{n=1}^{\infty}$ denote a sequence of data compression codes for $\boldsymbol{X}$. The *distortion inf-spectrum* $\underline{\lambda}_{(\boldsymbol{X},\boldsymbol{f}(\boldsymbol{X}))}(\theta)$ for $\boldsymbol{f}(\boldsymbol{X})$ is defined by

$$\underline{\lambda}_{(\boldsymbol{X},\boldsymbol{f}(\boldsymbol{X}))}(\theta) \stackrel{\Delta}{=} \liminf_{n\to\infty} \Pr\{\frac{1}{n}\rho_n(X^n, f_n(X^n)) \le \theta\}.$$

For any $1>\varepsilon>0$, the e-sup-distortion rate $\overline{\Lambda}_{\varepsilon}(\boldsymbol{X},\boldsymbol{f}(\boldsymbol{X}))$ is defined by

$$\overline{\Lambda}_{\varepsilon}(\boldsymbol{X},\boldsymbol{f}(\boldsymbol{X})) \stackrel{\Delta}{=} \sup\{\theta: \underline{\lambda}_{(\boldsymbol{X},\boldsymbol{f}(\boldsymbol{X}))}(\theta) \le \varepsilon\},$$

which is exactly the quantile of $\underline{\lambda}_{(\boldsymbol{X},\boldsymbol{f}(\boldsymbol{X}))}(\theta)$.

Definition 6.

Fix $D>0$ and $1>\varepsilon>0$. R is a $(1-\varepsilon)$-achievable data compression rate at distortion D for a source $\boldsymbol{X}$ if there exists a sequence of data compression codes $f_n(\cdot)$ with

$$\limsup_{n\to\infty} (1/n)\log\|f_n\| = R,$$

and $(1-\varepsilon)$-sup-distortion rate less than or equal to D:

$$\overline{\Lambda}_{1-\varepsilon}(\boldsymbol{X},\boldsymbol{f}(\boldsymbol{X})) \le D.$$

Note that stating that the code has $(1-\varepsilon)$-sup-distortion rate less than or equal to D is equivalent to stating that the limsup of the probability of excessived istortion (i.e., distortion larger than D) is smaller than ε: $\limsup_{n\to\infty} \Pr\{(1/n)\rho_n(X^n, f_n(X^n)) > D\} < \varepsilon$. The infimum $(1-\varepsilon)$-achievable data compression rate at distortion D for $\boldsymbol{X}$ is denoted by $T_{1-\varepsilon}(D, \boldsymbol{X})$.

Theorem 3. (General data compression theorem)

Fix $D>0$ and $1>\varepsilon>0$. Let $\boldsymbol{X}$ and $\{\rho_n(\cdot,\cdot)\}_{n\ge 1}$ be given. Then

$$R_{(1-\varepsilon)^-}(D) \le T_{1-\varepsilon}(D,\boldsymbol{X}) \le R_{1-\varepsilon}(D),$$

where

$$R_{1-\varepsilon}(D) \stackrel{\Delta}{=} \inf_{\{P_{Y|X}:\, \overline{\Lambda}_{1-\varepsilon}(\boldsymbol{X},Y)\le D\}} \overline{I}(\boldsymbol{X};Y),$$

and $P_{Y|X}=\{P_{Y^n|X^n}\}_{n=1}^{\infty}$ denotes a sequence of conditional distributions satisfying the constraint $\overline{\Lambda}_{1-\varepsilon}(X,Y)\le D$. In other words, $T_{1-\varepsilon}(D,X)=R_{1-\varepsilon}(D)$, except possibly at the points of discontinuities of $R_{1-\varepsilon}(D)$ (which are countable).

Proof:

1. *Forward part (achievability)*: Choose $\gamma>0$. We will prove the existence of a sequence of block codes with

$$\limsup_{n\to\infty}(1/n)\log|C_n|<R_{1-\varepsilon}(D)+2\gamma,$$

and

$$\overline{\Lambda}_{1-\varepsilon}(X,f(X))<D+\gamma.$$

Step 1: Let $P_{\tilde{Y}|X}$ be the distribution achieving $R_{1-\varepsilon}(D)$, and let $P_{\tilde{Y}}$ be the Y-marginal of $P_X P_{\tilde{Y}|X}$.

Step 2: Let R satisfy $R_{1-\varepsilon}(D)+2\gamma>R>R_{1-\varepsilon}(D)+\gamma$. Choose $M=e^{nR}$ n-blocks independently according to $P_{\tilde{Y}}$, and denote the resulting random set by C_n.

Step 3: For a given C_n, we denote by $A(C_n)$ the set of sequences $x^n\in\mathcal{X}^n$ such that there exists $y^n\in C_n$ with

$$(1/n)\rho_n(x^n,y^n)\le D+\gamma.$$

Step 4: *Claim:*

$$\limsup_{n\to\infty}E_{\tilde{Y}}[P_{X^n}(A^c(C_n))]<\varepsilon.$$

The proof of this claim is provided in Appendix A.

Therefore there exists (a sequence of) C_n^* such that

$$\limsup_{n\to\infty}P_{X^n}(A^c(C_n^*))<\varepsilon.$$

step 5: Define a sequence of codes $\{f_n\}$ by

$$f_n(x^n)=\begin{cases}\arg\min\limits_{y^n\in C_n^*}\rho_n(x^n,y^n), & \text{if } x^n\in A(C_n^*);\\ \underline{0} & \text{otherwise,}\end{cases}$$

where $\underline{0}$ is a fixed default n-tuple in $\mathcal{Y}^n$. Then

$$\{x^n\in\mathcal{X}^n:\frac{1}{n}\rho_n(x^n,f_n(x^n))\le D+\gamma\}\supset A(C_n^*),$$

since $(\forall x^n\in A(C_n^*))$ there exists $y^n\in C_n^*$ such that $(1/n)\rho_n(x^n,y^n)\le D+\gamma$, which by definition of f_n implies that $(1/n)\rho_n(x^n f_n(x^n))\le D+\gamma$.

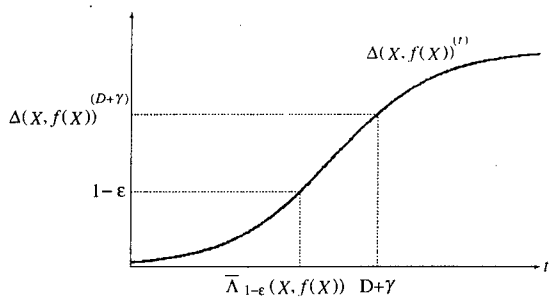


Fig. 2. $\underline{\lambda}_{(X,f(X))}(D+\gamma)>1-\varepsilon\Rightarrow\overline{\Lambda}_{1-\varepsilon}(X,f(X))<D+\gamma$.

Step 6: Consequently,

$$\begin{aligned}&\underline{\lambda}_{(X,f(X))}(D+\gamma)\\&=\liminf_{n\to\infty}P_{X^n}\{x^n\in\mathcal{X}^n:\frac{1}{n}\rho_n(x^n,f_n(x^n))\le D+\gamma\}\\&\ge\liminf_{n\to\infty}P_{X^n}\{A(C_n^*)\}\\&=1-\limsup_{n\to\infty}P_{X^n}\{A^c(C_n^*)\}\\&>1-\varepsilon.\end{aligned}$$

Hence,

$$\overline{\Lambda}_{1-\varepsilon}(X,f(X))<D+\gamma,$$

where the last step is clearly depicted in Fig. 2.

This proves the forward part.

2. *Converse part:* We show that for any sequence of encoders $\{f_n\}_{n=1}^{\infty}$, if

$$\overline{\Lambda}_{(1-\varepsilon)\text{-}}(X,f(X))\le D,$$

then

$$\limsup_{n\to\infty}(1/n)\log\|f_n\|\ge R_{(1-\varepsilon)\text{-}}(D).$$

Let

$$P_{\tilde{Y}^n|X^n}(y^n|x^n)\triangleq\begin{cases}1, & \text{if } y^n=f_n(x^n)\\ 0, & \text{otherwise.}\end{cases}$$

Then to evaluate the statistical properties of the random variable $(1/n)\rho_n(X^n,f_n(X^n))$ under distribution P_{X^n} is equivalent to evaluating the random variable $(1/n)\rho_n(X^n,\tilde{Y}^n)$ under distribution $P_{X^n\tilde{Y}^n}$. Therefore

$$R_{(1-\varepsilon)\text{-}}(D)=\inf_{\{P_{Y|X}:\overline{\Lambda}_{(1-\varepsilon)\text{-}}(X,Y)\le D\}}\overline{I}(X,Y)$$

$$\leq \overline{I}(X;\hat{Y})$$

$$\leq \overline{H}(\hat{Y}) - \underline{H}(\hat{Y}|X)$$

$$\leq \overline{H}(\hat{Y})$$

$$\leq \limsup_{n\to\infty} \frac{1}{n} \log \|f_n\|,$$

where the second inequality follows from [5, Lemma 3.2] (cf (3.12) with $\gamma=1^-$ and $\delta=0$), and the third inequality follows from the fact that $\underline{H}(\hat{Y}|X) \geq 0$.

■

Observations:

1. *Comparison with Steinberg and Verdú's result* [9]. If $\varepsilon \downarrow 0$, then

$$R_{(1-\varepsilon)}(D) \uparrow R_1(D) \stackrel{\Delta}{=} \inf_{\{P_{Y|X}: \overline{\Lambda}_1(X,Y) \leq D\}} \overline{I}(X;Y).$$

Remark that $R_1(D)$ is nothing but the *sup rate-distortion function* $\overline{R}(D)$ described in Definition 14 of [9]. Therefore, this theorem reduces to Theorem 10.(a) of [9] when $\varepsilon \downarrow 0$. Note that according to the terminology of [9, Definition 14], $R_{1-\varepsilon}(D)$ may be called the $(1-\varepsilon)$-*sup rate-distortion function*.

2. *Comparison with the data compaction theorem.* For the probability-of-error distortion measure $\rho_n: \mathcal{X}^n \to \mathcal{X}^n$, namely,

$$\rho_n(x^n, \hat{x}^n) = \begin{cases} n, & \text{if } x^n \neq \hat{x}^n; \\ 0, & \text{otherwise,} \end{cases}$$

we define a data compression code $f_n: \mathcal{X}^n \to \mathcal{X}^n$ based on a chosen data compaction code book $C_n \subset \mathcal{X}^n$:

$$f_n(x^n) = \begin{cases} x^n, & \text{if } x^n \in C_n; \\ \underline{0}, & \text{if } x^n \notin C_n, \end{cases}$$

where $\underline{0}$ is some default element in $\mathcal{X}^n$. Then $(1/n)\rho_n(x^n, f_n(x^n))$ is either 1 or 0 which results in a cumulative distribution function as shown in Fig. 3. Consequently, for any $\delta \in [0, 1)$,

$$\Pr\{\frac{1}{n}\rho_n(X^n, f_n(X^n)) \leq \delta\} = \Pr\{X^n = f_n(X^n)\}.$$

In other words, the condition

$$\overline{\Lambda}_{1-\varepsilon}(X, f(X)) \leq \delta$$

is equivalent to

$$\liminf_{n\to\infty} \Pr\{X^n = f_n(X^n)\} > 1 - \varepsilon,$$

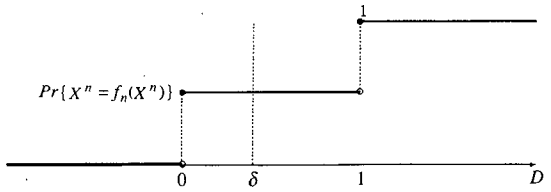


Fig. 3. The CDF of $(1/n)\,\rho_n(X^n, f_n(X^n))$ for the probability-of-error distortion measure.

which is exactly the same as $\limsup_{n\to\infty} \Pr\{X^n \neq f_n(X^n)\} < \varepsilon$.

By comparing the source compaction and compression theorems, we remark that $\overline{H}_{1-\varepsilon}(X)$ is indeed the counterpart of $R_{1-\varepsilon}(\delta)$ for the probability-of-error distortion measure and $\delta \in [0,1)$. In particular, in the extreme case where ε goes to zero,

$$\overline{H}(X) = \inf_{\{P_{\hat{X}|X}: \limsup_{n\to\infty} \Pr(X^n \neq \hat{X}^n) = 0\}} \overline{I}(X;\hat{X})$$

which follows from the fact that (cf (3.12) and (3.14) in [5, Lemma 3.2)

$$\overline{H}(X) - \overline{H}(X|\hat{X}) \leq \overline{I}(X;\hat{X}) \leq \overline{H}(X) - \underline{H}(X|\hat{X}),$$

and $\overline{H}(X|\hat{X}) = \underline{H}(X|\hat{X}) = 0$. Therefore, in this case, the data compression theorem reduces (as expected) to the data compaction theorem (Theorem 2).

III. NEYMAN-PEARSON HYPOTHESIS TESTING

In Neyman-Pearson hypothesis testing, the objective is to decide between two different explanations for the observed data. More specifically, given a sequence of observations with unknown underlying distribution Q, we consider two hypotheses:

- H_0: $Q = P_X$ (null hypothesis).
- H_1: $Q = P_{\hat{X}}$ (alternative hypothesis).

If we accept hypothesis H_1 when H_0 is actually true, we obtain what is known as a *type-I error*, and the probability of this event is denoted by α [2]. Accepting hypothesis H_0 when H_1 is actually true results in what we call a *type-II error*; the probability of this event is denoted by β. In general, the goal is to minimize *both* error probabilities; but there is a tradeoff since if α is reduced beyond a certain threshold then β increases (and vice-versa). Hence, we minimize one of the error probabilities subject to a constraint on the other error probability.

In the case of an arbitrary sequence of observations, the general expression of the Neyman-Pearson type-II error exponent subject to a constant bound on

the type-I error has been proved in [4, Theorem 1]. We re-formulate the expression in terms of the ε-inf/sup-divergence rates in the next theorem.

Theorem 4. (Neyman-Pearson type-II error exponent for a fixed test level)

Consider a sequence of random observations which is assumed to have a probability distribution governed by either P_X (null hypothesis) or $P_{\hat{X}}$ (alternative hypothesis). Then, the type-II error exponent satisfies

$$\overline{D}_{\varepsilon^-}(X \| \hat{X}) \le \limsup_{n\to\infty} -\frac{1}{n}\log \beta_n^*(\varepsilon) \le \overline{D}_{\varepsilon}(X \| \hat{X})$$

$$\underline{D}_{\varepsilon^-}(X \| \hat{X}) \le \liminf_{n\to\infty} -\frac{1}{n}\log \beta_n^*(\varepsilon) \le \underline{D}_{\varepsilon}(X \| \hat{X})$$

where $\beta_n^*(\varepsilon)$ represents the minimum type-II error probability subject to a fixed type-I error bound $\varepsilon \in [0,1)$.

The general formula for Neyman-Pearson type-II error exponent subject to an exponential test level is also proved in [4, Theorem 3]. We, herein provide an extension of this result and express it in terms of the ε-inf/sup-divergence rates.

Theorem 5. (Neyman-Pearson type-II error exponent for an exponential test level)

Fix $s \in (0,1)$ and $\varepsilon \in [0,1)$. It is possible to choose decision regions for a binary hypothesis testing problem with arbitrary datawords of blocklength n, (which are governed by either the null hypothesis distribution P_X or the alternative hypothesis distribution $P_{\hat{X}}$), such that

$$\liminf_{n\to\infty} -\frac{1}{n}\log \beta_n^* \ge \overline{D}_{\varepsilon}(\hat{X}^{(s)} \| \hat{X}) \text{ and}$$

$$\limsup_{n\to\infty} -\frac{1}{n}\log \alpha_n \ge \underline{D}_{1-\varepsilon}(\hat{X}^{(s)} \| X), \tag{9}$$

or

$$\liminf_{n\to\infty} -\frac{1}{n}\log \beta_n \ge \underline{D}_{\varepsilon}(\hat{X}^{(s)} \| \hat{X}) \text{ and}$$

$$\limsup_{n\to\infty} -\frac{1}{n}\log \alpha_n \ge \overline{D}_{1-\varepsilon}(\hat{X}^{(s)} \| X), \tag{10}$$

where $\hat{X}^{(s)}$ exhibits the tilted distributions $\{P_{\hat{X}^n}^{(s)}\}_{n=1}^{\infty}$ defined by

$$dP_{\hat{X}^n}^{(s)}(x^n) \triangleq \frac{1}{\Omega_n(s)} \exp\{s \log \frac{dP_{X^n}}{dP_{\hat{X}^n}}(x^n)\}\, dP_{\hat{X}^n}(x^n),$$

and

$$\Omega_n(s) \triangleq \int_{\mathcal{X}^n} \exp\{s \log \frac{dP_{X^n}}{dP_{\hat{X}^n}}(x^n)\} dP_{\hat{X}^n}(x^n).$$

Here, α_n and β_n are the type-I and type-II error probabilities respectively.

Proof:

For ease of notation, we use $\tilde{X}$ to represent $\hat{X}^{(s)}$. We only prove (9); (10) can be similarly demonstrated.

By definition of $dP_{\hat{X}^n}^{(s)}(\cdot)$, we have

$$\frac{1}{s}[\frac{1}{n} d_{\tilde{X}^n}(\tilde{X}^n \| \hat{X}^n)] + \frac{1}{1-s}[\frac{1}{n} d_{\tilde{X}^n}(\tilde{X}^n \| \hat{X}^n)]$$

$$= -\frac{1}{s(1-s)}[\frac{1}{n}\log \Omega_n(s)]. \tag{11}$$

Let $\overline{\Omega} \triangleq \limsup_{n\to\infty} (1/n)\log\Omega_n(s)$. Then, for any $\gamma > 0$, $\exists N_0$ such that $\forall n > N_0$,

$$(1/n)\log\Omega_n(s) < \overline{\Omega} + \gamma.$$

From (11),

$$\underline{d}_{\tilde{X}^n \| \hat{X}^n}(\theta) = \liminf_{n\to\infty} \Pr\{\frac{1}{n} d_{\tilde{X}^n}(\tilde{X}^n \| X^n) \le \theta\}$$

$$= \liminf_{n\to\infty} \Pr\{-\frac{1}{1-s}[\frac{1}{n} d_{\tilde{X}^n}(\tilde{X}^n \| X^n)] - \frac{1}{s(1-s)}[\frac{1}{n}\log\Omega_n(s)] \le \frac{\theta}{s}\}$$

$$\le \frac{\theta}{s}\}$$

$$= \liminf_{n\to\infty} \Pr\{\frac{1}{n} d_{\tilde{X}^n}(\tilde{X}^n \| X^n) \ge -\frac{1-s}{s}\theta - \frac{1}{s}[\frac{1}{n}\log\Omega_n(s)]\}$$

$$\le \liminf_{n\to\infty} \Pr\{\frac{1}{n} d_{\tilde{X}^n}(\tilde{X}^n \| X^n) > -\frac{1-s}{s}\theta - \frac{1}{s}\overline{\Omega} - \frac{\gamma}{s}\}$$

$$= 1 - \limsup_{n\to\infty} \Pr\{\frac{1}{n} d_{\tilde{X}^n}(\tilde{X}^n \| X^n) \le -\frac{1-s}{s}\theta - \frac{1}{s}\overline{\Omega} - \frac{\gamma}{s}\}$$

$$= 1 - d_{\tilde{X}^n \| \hat{X}^n}(-\frac{1-s}{s}\theta - \frac{1}{s}\overline{\Omega} - \frac{\gamma}{s}).$$

Thus,

$$\overline{D}_{\varepsilon}(\tilde{X} \| \hat{X}) = \sup\{\theta: \underline{d}_{\tilde{X}^n \| \hat{X}^n}(\theta) \le \varepsilon\}$$

$$\ge \sup\{\theta: 1 - \overline{d}_{\tilde{X}^n \| \hat{X}^n}(-\frac{1-s}{s}\theta - \frac{1}{s}(\overline{\Omega} + \gamma)) < \varepsilon\}$$

$$= \sup\{-\frac{1}{1-s}(\overline{\Omega} + \gamma) - \frac{s}{1-s}\theta' : \underline{d}_{\tilde{X}^n \| \hat{X}^n}(\theta') > 1 - \varepsilon\}$$

$$= -\frac{1}{1-s}(\overline{\Omega} + \gamma) - \frac{s}{1-s}\inf\{\theta' : \underline{d}_{\tilde{X}^n \| \hat{X}^n}(\theta') > 1 - \varepsilon\}$$

$$= -\frac{1}{1-s}(\overline{\Omega} + \gamma) - \frac{s}{1-s}\sup\{\theta' : \underline{d}_{\tilde{X}^n \| \hat{X}^n}(\theta') \le 1 - \varepsilon\}$$

$$= -\frac{1}{1-s}(\overline{\Omega} + \gamma) - \frac{s}{1-s}\underline{D}_{1-\varepsilon}(\tilde{X} \| X).$$

Finally, choose the acceptance region for null hypothesis as

$$\{\frac{1}{n}\log\frac{dP_{\tilde{X}^n}}{dP_{\hat{X}^n}}(X^n)\geq\overline{D}_\varepsilon(\tilde{X}\|\hat{X})\}.$$

Therefore

$$\beta_n = P_{\hat{X}^n}\{\frac{1}{n}\log\frac{dP_{\tilde{X}^n}}{dP_{\hat{X}^n}}(X^n)\geq\overline{D}_\varepsilon(\tilde{X}\|\hat{X})\}$$

$$\leq\exp\{-n\,\overline{D}_\varepsilon(\tilde{X}\|\hat{X})\},$$

and

$$\alpha_n = P_{X^n}\{\frac{1}{n}\log\frac{dP_{\tilde{X}^n}}{dP_{\hat{X}^n}}(X^n)<\overline{D}_\varepsilon(\tilde{X}\|\hat{X})\}$$

$$\leq P_{X^n}\{\frac{1}{n}\log\frac{dP_{\tilde{X}^n}}{dP_{\hat{X}^n}}(X^n)<-\frac{1}{1-s}(\overline{\Omega}+\gamma)-\frac{s}{1-s}\underline{D}_{1-\varepsilon}(\tilde{X}\|X)\}$$

$$=P_{X^n}\{-\frac{1}{1-s}[\frac{1}{n}\log\frac{dP_{\tilde{X}^n}}{dP_{\hat{X}^n}}(X^n)]-\frac{1}{s(1-s)}[\frac{1}{n}\log\Omega_n(s)]$$

$$<-\frac{\overline{\Omega}+\gamma}{s(1-s)}-\frac{1}{1-s}\underline{D}_{1-\varepsilon}(\tilde{X}\|X)\}$$

$$=P_{X^n}[\frac{1}{n}\log\frac{dP_{\tilde{X}^n}}{dP_{\hat{X}^n}}(X^n)]>\underline{D}_{1-\varepsilon}(\tilde{X}\|X)+\frac{1}{s}[\overline{\Omega}-\frac{1}{n}\log\Omega_n(s)]+\frac{\gamma}{s}\}.$$

Then, for $n>N_0$,

$$\alpha_n\leq P_{X^n}\{\frac{1}{n}\log\frac{dP_{\tilde{X}^n}}{dP_{X^n}}(X^n)>\underline{D}_{1-\varepsilon}(\tilde{X}\|\hat{X})\}$$

$$\leq\exp\{-n\,\underline{D}_{1-\varepsilon}(\tilde{X}\|X)\}.$$

Consequently,

$$\liminf_{n\to\infty}-\frac{1}{n}\log\beta_n\geq\overline{D}_\varepsilon(\hat{X}^{(s)}\|\hat{X})\text{ and}$$

$$\liminf_{n\to\infty}-\frac{1}{n}\log\alpha_n\geq\underline{D}_{1-\varepsilon}(\hat{X}^{(s)}\|X).$$

■

IV. CONCLUSIONS

In light of the new information quantiles introduced in [3], a generalized version of the Asymptotic Equipartition Property (AEP) is proved. General data compaction and compression (source coding) theorems for block codes and general expressions for the Neyman-Pearson hypothesis testing type-II error exponent are also derived.

Finally, it is demonstrated that by using these new quantities, Shannon's coding theorems can be reformulated in their *most general form* and the error probability of an *arbitrary* stochastic communication system can be determined.

ACKNOWLEDGMENT

The authors would like to thank Prof. S. Verdú for his valuable advice and constructive criticism which helped improve the paper.

NOMENCLATURE

$T_{1-\varepsilon}(X)$	$(1-\varepsilon)$-achievable data compaction rate
$T_{1-\varepsilon}(D,X)$	$(1-\varepsilon)$-achievable data compression rate at distortion D
$\underline{D}_\delta(X\|\|\hat{X})$	δ-inf-divergence rate
$\underline{H}_\delta(X)$	δ-inf-entropy rate
$\underline{I}_\delta(X;Y)$	δ-inf-information rate
$\overline{D}_\delta(X\|\|\hat{X})$	δ-sup-divergence rate
$\overline{H}_\delta(X)$	δ-sup-entropy rate
$\overline{I}_\delta(X;Y)$	δ-sup-information rate
$\overline{\Lambda}_\varepsilon(X;Y)$	ε-sup-distortion rate
C_ε	ε-capacity
C	channel capacity
$P_{W^n}=P_{Y^n\|X^n}$	channel transition distribution
$\underline{\lambda}_{(X,f(X))}(\theta)$	distortion inf-spectrum
$\underline{d}_{X\|\hat{X}}(\theta)$	divergence inf-spectrum
$\overline{d}_{X\|\hat{X}}(\theta)$	divergence sup-spectrum
$h_{X^n}(X^n)$	entropy density
$\underline{h}_X(\theta)$	entropy inf-Spectrum
$\overline{h}_X(\theta)$	entropy sup-Spectrum
$\underline{D}_\delta(X\|\|\hat{X})$	inf-divergence rate
$\underline{H}(X)$	inf-entropy rate
$\underline{I}(X;Y)$	inf-information rate
$i_{X^nW^n}(X^n;Y^n)$	information density
$\underline{i}_{(X,Y)}(\theta)$	information inf-spectrum
$\overline{i}_{(X,Y)}(\theta)$	information sup-spectrum
$\mathcal{A}$	input alphabet
P_{X^n}	input distributions
$d_{X^n}(X^n\|\|\hat{X}^n)$	log-likelihood ratio
$\mathcal{B}$	output alphabet
$\overline{D}(X\|\|\hat{X})$	sup-divergence rate
$\overline{H}(X)$	sup-entropy rate
$\overline{I}(X;Y)$	sup-information rate

REFERENCES

1. Alajaji F. and T. Fuja. "A Communication Channel Modeled on Contagion," *IEEE Transactions on Information Theory*, IT-40(6):2035-2041, November (1994).
2. Blahut, R.E. *Principles and Practice of Information Theory*, Addison Wesley, Massachusetts (1988).
3. Chen P.-N. and F. Alajaji, "Generalized Source Coding Theorems and Hypothesis Testing: Part I - information Measures," *Journal of the Chinese Institute of Engineers*, Vol. 21, No. 3, May (1998).

4. Che, P.-N. "General Formulas for the Neyman-Pearson type-II Error Exponent Subject to Fixed and Exponential type-I Error Bounds," *IEEE Transactions on Information Theory*, IT-42(1):316-323, January (1996).
5. Cover T.M. and J.A. *Thomas, Elements of Information Theory*, Wiley, New York (1991).
6. Csiszr, I. "Information theory and Ergodic Theory," *Problems of Control and Inform.* Theory, 16(1):3-27 (1987).
7. Gray, R.M. *Entropy and Information Theory*. Springer-Verlag, New York (1990).
8. Han T.S. and S. Verdú, "Approximation Theory of Output Statistics," *IEEE Transactions on Information Theory*, IT-39(3):752-772, May (1993).
9. Steinberg Y. and S. Verdú, "Simulation of Random Processes and Rate-distortion Theory," *IEEE Transactions on Information Theory*, IT-42(1):63-86, Jan. (1996).
10. Vembu, S.S. Verdú and Y. Steinberg, "The Source-channel Separation Theorem Revisited," *IEEE Transactions on Information Theory*, IT-41(1):44-54, Jan. (1995).
11. Verd S. and T.S. Han, "A General Formula for Channel Capacity," *IEEE Transactions on Information Theory*, IT-40(4):1147-1157, Jul. (1994).

APPENDIX A

Claim (*cf Proof of Theorem 3*)

$$\limsup_{n\to\infty} E_{\tilde{Y}}[P_{X^n}(A^c(C_n^*))] < \varepsilon .$$

***Proof*:**

step 1: Define

$$A_{n,\gamma}^{(\varepsilon)} \triangleq \{(x^n, y^n): \frac{1}{n}\rho_n(x^n, y^n) \le \overline{\Lambda}_{1-\varepsilon}(X,\tilde{Y}) + \gamma,$$

$$\frac{1}{n} i_{X^nY^n}(x^n, y^n) \le \overline{I}(X,\tilde{Y}) + \gamma\} .$$

Since

$$\liminf_{n\to\infty} \Pr(D \triangleq \{(1/n)\rho_n(X^n, \tilde{Y}^n) \le \overline{\Lambda}_{1-\varepsilon}(X,\tilde{Y}) + \gamma\}) > 1-\varepsilon ,$$

and

$$\liminf_{n\to\infty} \Pr(E \triangleq \{(1/n) i_{X^n\tilde{Y}^n}(X^n, \tilde{Y}^n) \le \overline{I}(X,\tilde{Y}) + \gamma\}) = 1 ,$$

we have

$$\liminf_{n\to\infty} \Pr(A_{n,\gamma}^{(\varepsilon)}) = \liminf_{n\to\infty} \Pr(D \cap E)$$

$$\ge \liminf_{n\to\infty} \Pr(D) + \liminf_{n\to\infty} \Pr(E) - 1$$

$$> (1-\varepsilon) + 1 - 1 = 1 - \varepsilon .$$

step 2: Let $K(x^n, y^n)$ be the indicator function of $A_{n,\gamma}^{(\varepsilon)}$:

$$K(x^n, y^n) = \begin{cases} 1, & \text{if } (x^n, y^n) \in A_{n,\gamma}^{(\varepsilon)}; \\ 0, & \text{otherwise.} \end{cases}$$

step 3: By following a similar argument in [9, equations (9)-(12)], we obtain,

$$E_{\tilde{Y}}[P_{X^n}(A^c(C_n^*))]$$

$$= \sum_{C_n^*} P_{\tilde{Y}^n}(C_n^*) \sum_{x^n \notin A(C_n^*)} P_{X^n}(x^n)$$

$$= \sum_{x^n \in \mathcal{X}^n} P_{X^n}(x^n) \sum_{C_n^*: x^n \notin A(C_n^*)} P_{\tilde{Y}^n}(C_n^*)$$

$$= \sum_{x^n \in \mathcal{X}^n} P_{X^n}(x^n)\left(1 - \sum_{y^n \in \mathcal{Y}^n} P_{\tilde{Y}^n}(y^n) K(x^n, y^n)\right)^M$$

$$\le \sum_{x^n \in \mathcal{X}^n} P_{X^n}(x^n)\left(1 - e^{-n(\overline{I}(X;\tilde{Y})+\gamma)} \times \sum_{y^n \in \mathcal{Y}^n} P_{\tilde{Y}^n|X^n}(y^n \mid x^n) K(x^n, y^n)\right)^M$$

$$\le 1 - \sum_{x^n \in \mathcal{X}^n} \sum_{x^n \in I^n} P_{X^n}(x^n) P_{\tilde{Y}^n|X^n}(y^n \mid x^n) K(x^n, y^n)$$

$$+ \exp\{-e^{-n(R - R_{1-\varepsilon}(D)-\gamma)}\} .$$

Therefore

$$\limsup_{n\to\infty} E_{Y^n}[P_{X^n}(A^c(C_n^*))] \le 1 - \liminf_{n\to\infty} \Pr(A_{n,\gamma}^{(\varepsilon)})$$

$$< 1 - (1-\varepsilon) = \varepsilon .$$

■

APPENDIX B

Claim: Fix $\varepsilon \in [0, 1)$. $T_\varepsilon(X)$ is right-continuous in ε.

proof: Suppose $T_\varepsilon(X)$ is not right-continuous for some $\varepsilon \in [0, 1)$. Then there exists $\gamma > 0$ such that

$$T_{\varepsilon+\delta}(X) < T_\varepsilon(X) + 3\gamma \text{ for every } 1-\varepsilon > \delta > 0,$$

which guarantees the existence of R satisfying

$$T_{\varepsilon+\delta}(X) < R - \gamma < R < T_\varepsilon(X)$$

for every $1-\varepsilon > \delta > 0$. Hence, $R-\gamma$ is $(\varepsilon+\delta)$-achievable for every $1-\varepsilon > \delta > 0$, and R is not ε-achievable.

By definition of $(\varepsilon+\delta)$-achievability, there exists a code $D_n(\delta)$ such that

$$\limsup_{n\to\infty} (1/n)\log|D_n(\delta)| = R-\gamma \text{ and}$$

$$\limsup_{n\to\infty} Pe(D_n(\delta)) \le \varepsilon + \delta.$$

Therefore, there exists $M(\delta)$ such that for $n>M(\delta)$,

$$(1/n)\log|D_n(\delta)|<R \text{ and } Pe(D_n(\delta))<\varepsilon+2\delta.$$

Observe that if we increase the code size of $D_n(\delta)$ to obtain a new code $D'_n(\delta)$ with $(1/n)\log|D'_n(\delta)|=R$ for $n>M(\delta)$, then the error probability will not increase, i.e.,

$$Pe(D'_n(\delta))<\varepsilon+2\delta.$$

Now define a new code E_n as follows:

$$E_n=D'_n(\delta) \text{ for } M(\delta)<n\le\max\{M(\delta), M(\delta/2)\}$$

$$E_n=D'_n(\delta/2) \text{ for } \max\{M(\delta), M(\delta/2)\}<n \le\max\{M(\delta), M(\delta/2), M(\delta/3)\}$$

$$E_n=D'_n(\delta/3) \text{ for } \max\{M(\delta), M(\delta/2), M(\delta/3)\}<n \le\max\{M(\delta), M(\delta/2), \mathrm{M}(\delta/3), M(\delta/4)\}$$

$$\vdots$$

Then for $n>M(\delta)$, $(1/n)\log|E_n|=R$ but $\limsup_{n\to\infty} Pe(E_n)\le\varepsilon$, contradicting the fact that R is not e-achievable. ■

Claim:

$$T_{1-\varepsilon}(X)=\overline{H}_{\varepsilon}(X) \text{ for } \varepsilon\in(0,1] \text{ and } T_1(X)=0.$$

Proof:

The first result is an immediate consequence of the right-continuity of $T_{1-\varepsilon}(X)$ w.r.t. $(1-\varepsilon)\in[0,1)$. $T_1(X)$, by definition, is the infimum of the 1-achievable data compaction rate which requires the existence of codes C_n with

$$\limsup_{n\to\infty}(1/n)\log|C_n|=R,$$

and

$$\limsup_{n\to\infty} Pe(C_n)\le 1,$$

We can then choose an empty code set, and obtain $T_1(X)=0$.

Discussions of this paper may appear in the discussion section of a future issue. All discussions should be submitted to the Editor-in-Chief.

Manuscript Received: June 25, 1997
Revision Received: Jan. 11,1998
and Accepted: Jan. 24, 1998

來源編碼定理與檢定測試的一般定理：第二部份－操作極限

陳伯寧

國立交通大學電信工程系所

Fady Alajaji

Queen's University
Kingston, ON K7L 3N6, Canada

摘 要

從本論文第一部份的訊息量度觀點，我們證明了近似平均分割性質的衍申擴展性質，同時也建立了位意限定字碼數目來源的定長資料潔簡編碼定理與定長資料壓縮編碼定理。在論文的最後，我們也將分析固定第一類錯誤率上界的紐門皮爾森檢定測試第二類錯誤率指數等級之一般公式。

關鍵詞：近似平均分割性質、來源編碼定理、檢定測試、紐門皮爾森錯誤指數。

INTERFERMETRIC FIBER SENSORS BASED ON TRIANGULAR PHASE MODULATION

*Ching-Ting Lee, Lih-Wuu Chang
*Institute of Optical Sciences
National Central University
Chung-Li, Taiwan 320, R.O.C.

Pie-Yau Chien
Material Research Center
Chung-Shan Institute of Science and Technology
Tao-Yuan, Taiwan 325, R.O.C.

ABSTRACT

This work presents three novel signal processing methods capable of accurately detecting the optical phase delay in optical interferometric sensors. In the proposed methods, a triangular waveform modulation signal is used to modulate the optical phase of an optical interferometer for measuring optical phase delay. The optical phase delay in the first method is obtained by integrating the interferometric output signals with the gated-in signal. The optical phase delay in the second method can be measured from the time delay difference between the interferometric signals during the gated-in period. For the third method, the pseudo-heterodyne technique is applied, and the optical phase delay is measured from the phase difference between two output signals. Moreover, a fiber Sagnac interferometric rotation sensor is adopted to demonstrate the effectiveness of the proposed methods. Furthermore, experimental results confirm that the proposed methods have an extreme degree of sensitivity and good linearity.

Key Words: superluminescent diode; fiber optic gyroscope; phase-locked loop; E-O phase modulator.

I. INTRODUCTION

Fiber optic interferometric sensors have been widely investigated owing to their inherent advantages compared with conventional sensors. Those advantages include immunity to electromagnetic influence, high sensitivity, large dynamic range, long-distance remote operation ability and multiplexing convenience [5, 9, 10]. Applying the fiber interferometric sensors allows for precise measurement of several physical parameters. Usually, the output of an optical interferometric sensor is a cosinusoidal function related to the optical phase delay generated from optical waves guided in the two arms of an interferometer. In addition, the environment temperature fluctuation causes a drifting of the phase bias point, thereby necessitating a quadrature phase stabilization servo loop to ensure that the output characteristics are within a linear region [11].

Operating an interferometer with high sensitivity and good linear scale factor involves modulating the optical phase by an external carrier signal to shift

*Correspondence addressee

it from a low frequency band to a high one. Therefore, the optical phase delay can be detected from the amplitude, phase or time delay of the carrier signal. For the phase measurement of an interferometer, the sinusoidal, sawtooth, square and triangular modulation waveforms have been reported [3, 6, 13, 15, 16, 17, 18, 19, 20, 21]. Furthermore, the time and frequency multiplexing schemes, by using these modulation techniques, have also been presented [2, 12, 14]. For a ramp phase modulation, a sawtooth waveform with peak phase deviation of 2π is used to generate a pure sinusoidal carrier signal [22]; the optical phase delay is measured from the phase difference between the sawtooth and sinusoidal carrier signals. Meanwhile, the dual ramp modulation has also been applied to the closed-loop fiber optic gyroscope (FOG), in which a feedback signal is applied to null the rotation rate [1]. In that method, the duty cycle of the triangular waveform is adopted as a feedback factor in a closed-loop FOG [1]; the rotation rate can be derived by altering the duty cycle of the triangular modulation signal. A previous work developed the deep phase modulation with triangular waveform of a Mach-Zehnder interferometer, attaining a scanning path length exceeding 10cm [7]. It was constructed on an all fiber optical spectrum analyzer using fast Fourier transform (FFT). Moreover, the wavelength of light returned from fiber Bragg grating could be distinguished easily. The triangular modulation waveform can also be used to implement the nulling operation of a Sagnac interferometer [4]. In general, the triangular modulation signal holds the following advantages over sinusoidal, sawtooth and square modulation signals: less harmonic frequency components in the output signal of an optical interferometer and more easily implemented since it uses a conventional digital up-down counter followed by a digital to analog (D/A) converter.

In this work, we present three optical phase measurement methods with open-loop operation based on the triangular waveform modulation. For the first method, i.e. the phase sensitive detection method (PSDM), the optical phase delay is transferred into the amplitude of a carrier signal, and a synchronized mixing circuit is adopted to demodulate the signal. For the second method, i.e. the direct phase difference method (DPDM), a gating signal is generated from the triangular modulation signal and the optical phase delay is measured from the time delay between the gated-in signals. For the third method, i.e. the pseudo-heterodyne detection method (PHDM), a carrier signal is also generated from the triangular modulation signal and the optical phase delay is measured from the phase difference between two carrier signals.

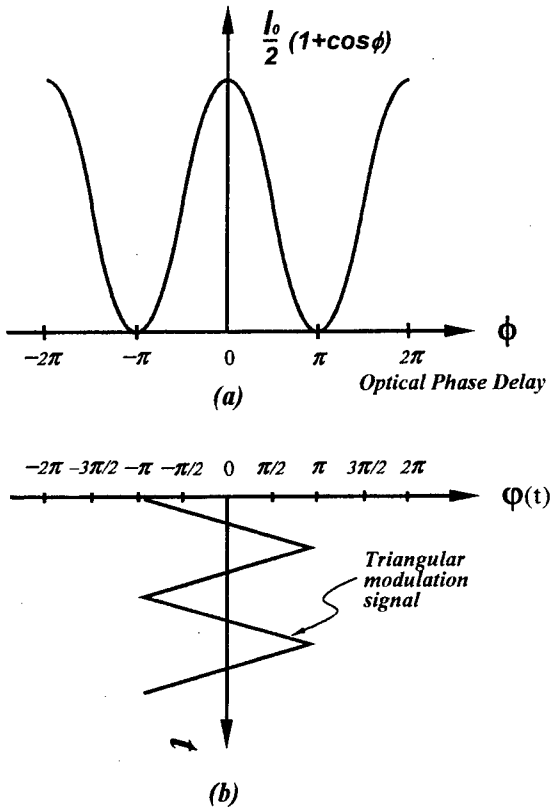


Fig. 1. (a) Transmission factor of a fiber optic interferometric sensor. (b) applied triangular modulation signal.

II. THEORETICAL ANALYSIS AND EXPERIMENTAL RESULTS

For a fiber optic interferometric sensor, its transmission factor, defined as the dependence of the detected intensity on the optical phase delay, is shown in Fig.1(a) [8]. The optical phase is modulated by a triangular waveform (Fig.1(b)), and the effective peak-to-peak phase modulation index is adjusted to 2π. The triangular modulation signal can be expressed as

$$S_+=\alpha T/2+\alpha t, \quad \text{for } -T/2<t<0, \tag{1a}$$

and

$$S_-=\alpha T/2-\alpha t, \quad \text{for } 0<t<T/2, \tag{1b}$$

where α and T denote the slope and period of the triangular modulation signal, respectively. The output signal of the optical interferometric sensor with an optical phase delay ϕ after applying triangular modulation signal can be expressed as

$$I_+=I_o/2[1+\cos(\alpha T/2+\alpha t+\phi)],\ \text{for}\ -T/2<t<0,\quad(2a)$$

and

$$I_-=I_o/2[1+\cos(\alpha T/2-\alpha t-\phi)],\ \text{for}\ 0<t<T/2,\quad(2b)$$

where I_o and $\alpha T/2$ represent the optical power of the light source and the peak phase modulation index, respectively. Optical phase delay ϕ equals $2\pi n\nu L/c$, where ν denotes the frequency of the light source, n and L are the refractive index and the optical path difference, respectively. However, when $\alpha T/2$ is selected to be 2π, (2) can be rewritten as

$$I_+=I_o/2[1+\cos(2\omega_m t+\phi)],\ \text{for}\ -T/2<t<0.\quad(3a)$$

and

$$I_-=I_o/2[1+\cos(2\omega_m t-\phi)],\ \text{for}\ 0<t<T/2,\quad(3b)$$

where $\omega_m=2\pi/T$ is the angular frequency of the triangular modulation signal. Applying a triangular modulation signal to optical interferometers allows us to derive a pure sinusoidal carrier signal with carrier frequency $2\omega_m$ for each half period $T/2$. Meanwhile, the optical phase delay is transferred into the phase of the carrier signal. The phase of the two carrier signals is in the opposite direction.

To obtain a stable output of an optical interferometric sensor, a zero path length difference Sangac interferometer is adopted for the principle demonstration. A superluminescent diode (SLD), a polarization maintaining fiber coupler, a multi-function integrated-optic chip (include a coupler, a polarizer and two electrooptical (E-O) phase modulators), and a polarization maintaining fiber are used to construct a Sangac interferometer system. Fig. 2 depicts a Sangac interferometer followed by different signal processing units. The triangular modulation signal generated by a function generator is employed to modulate the electrooptical phase modulator 1 (E-O1) without any distortion. Although the optical phase delay can be determined from the rotational rate of the Sangac interferometer, obtaining a large delay when the Sangac interferometer operates at a high rotational rate is extremely difficult. Thus, the induced optical phase bias can be easily generated by varying the frequency of the sawtooth modulation signal employed at electrooptical phase modulator 2 (E-O2). Meanwhile, the phase modulation index is set at 2π.

When the frequency of the triangular modulation signal is set at the characteristic frequency $f_c=1/2\tau$, where τ denotes the propagation time of the light in the fiber loop of the Sagnac interferometer, the optical modulation phase $\phi(t)$ of the counterclockwise wave is opposite to the modulation phase $\phi(t-\tau)$ of the clockwise wave. The effective modulation phase $\phi_{eff}(t)$ can be written as

$$\phi_{eff}(t)=\phi(t)-\phi(t-\tau)=2\phi(t).\quad(4)$$

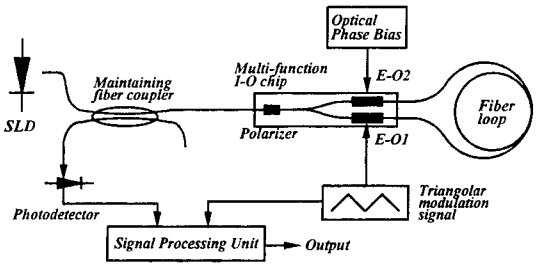


Fig. 2. Block diagram of a triangular phase modulated Sagnac interferometer with associated signal processing units.

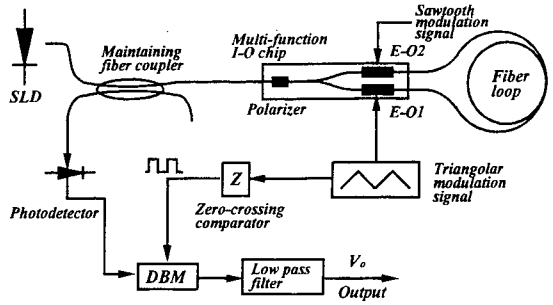


Fig. 3. Experimental setup of Sagnac interferometer system for phase sensitive detection method. SLD: superluminescent diode, DBM: double-balanced mixer, E-O: electrooptical phase modulator.

The operating conditions of the Sagnac interferometer, shown in Fig. 3, Fig. 9 and Fig. 12, are all the same for the three methods. A superluminescent diode (SLD) with wavelength $\lambda=1.3\mu$m is used as the light source to reduce the phase noise. The driving current and the temperature of the SLD is stabilized at $\Delta I/I\leq10^{-5}$ and $\Delta T\leq0.01$°C, respectively. The polarization maintaining fiber with a length of 700 m is coiled to a bobbin with a diameter of 12 cm. In addition, the E-O1 and E-O2 are used for phase modulation and phase bias generation of the Sagnac interferometer, respectively. Next, the frequency of the triangular modulation signal is selected to the characteristic frequency 144 kHz. The clockwise and the counterclockwise lights are mixed together through a fiber coupler and, then, received using a photodetector. The information of the optical phase delay can be obtained using three signal processing methods. Those methods are explained in the following.

1. Phase sensitive detection method

Figure 3 presents the experimental setup of the phase sensitive detection method (PSDM). In this figure, triangular and sawtooth modulation signals are separately applied to modulate an individual E-O phase modulator of the Sangac interferometer. The waveform of the triangular modulation signal is shaped into a square waveform by passing through an electronic zero-crossing comparator. The square signal is then used as a referenced gating signal for the double-balanced mixer (DBM). The output signal of the optical interferometric sensor is detected using a photodetector. Next, the detected signal and referenced gating signal are applied to the signal and reference input ports of the DBM, respectively. The DBM can generate a positive (+1) and a negative (-1) gain within the front and rear half periods of the gating signal, respectively. Figs. 4(a), 4(b) and 4(c) display the photodetector output signal, gating square signal and DBM output signal, respectively, under optical phase delay $\phi=-\pi/2$ and $\pi/2$ rad. The DBM output is passed through a low pass filter and the averaged output signal V_o can be expressed as

$$V_o = \frac{-I_0}{2}\frac{1}{T}\left[\int_0^{T/4}\cos(2\omega_m t+\phi)\,dt+\int_{T/4}^{T/2}\cos(2\omega_m t-\phi)\,dt - \int_{T/2}^{3T/4}\cos(2\omega_m t-\phi)\,dt-\int_{3T/4}^{T}\cos(2\omega_m t+\phi)\,dt\right]$$

$$= \frac{I_0}{2}\frac{-1}{2\omega_m T}\left[\sin(2\omega_m t+\phi)\Big|_0^{T/4}+\sin(2\omega_m t-\phi)\Big|_{T/4}^{T/2} - \sin(2\omega_m t-\phi)\Big|_{T/2}^{3T/4}-\sin(2\omega_m t+\phi)\Big|_{3T/4}^{T}\right]$$

$$= \frac{I_o}{2}\frac{8}{\pi}\sin\phi, \qquad (5)$$

where $4I_o/\pi$ denotes the transfer gain of the phase sensitive detection of the DBM. The output form in Eq. (5) is the same as that of the conventional lock-in demodulation (LID) technique at a fundamental frequency. When optical interferometers are modulated with a sinusoidal modulation signal, many high order harmonic frequencies are generated due to the nonlinear scanning characteristic of the sinusoidal waveform. However, the output signal is a pure sinusoidal signal after a triangular modulation. In doing so, there is no requirement for any high Q-value band pass filter to eliminate the unwanted signal. The output signal can be directly processed in the PSDM. Therefore, the sensitivity and stability of the PSDM is superior to the conventional LID technique.

The characteristics and performance of the PSDM can be summarized as follows:

(a) Sensitivity: The sensitivity of the PSDM is $\phi_{min}\cong 1.0\times10^{-6}\ rad/Hz^{1/2}$. It is the same as that of the conventional LID technique used in an optical interferometric sensor;

(b) Dynamic range: The maximum detectable optical phase delay is $\phi_{max}=\pm\pi/2\ rad$. To obtain good linearity, ϕ_{max} is limited to $\phi_{max}\leq\pm\pi/10$, and $\sin\phi\cong\phi$;

(c) The output curve of PSDM is sinusoidal; and

(d) Temperature dependence: There is no requirement for a band pass filter in the signal processing unit. Therefore, the temperature dependent phase bias generated from the phase drift in the electronic circuit is less than 5 $\mu rad/°C$;

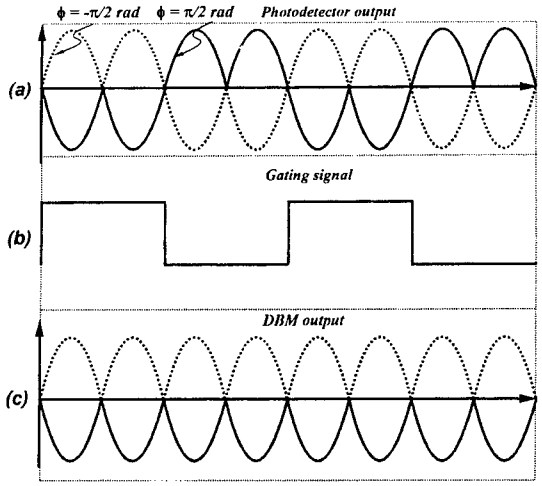
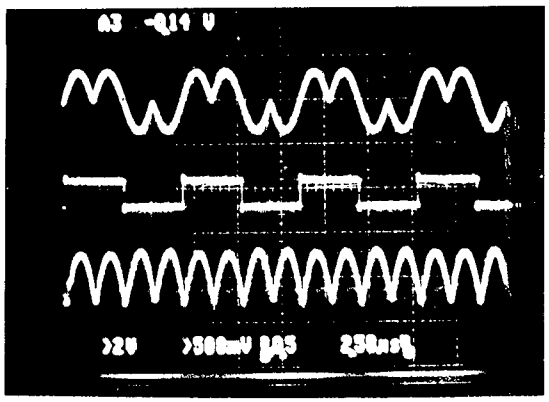


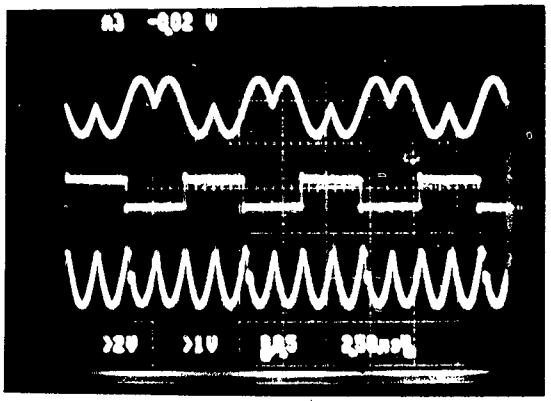
Fig. 4. Theoretical results of Fig. 3 under optical phase delay $-\pi/2$ rad and $\pi/2$ rad. (a) output signal of interferometer under triangular modulation, (b) gating signal , and (c) output signal of DBM.

As Fig. 3 indicates, a square gating signal with frequency 144 kHz, which was the same as that of the triangular modulation signal, was used to switch the output signal of the optical interferometer. When the output signal of the DBM was passed through a low pass filter with RC time constant 0.1sec, the optical phase delay was then directly measured. Figs. 5(a) and (b) summarize the experimental results of the DBM under optical phase delay $\phi=-\pi/2$ and $\pi/2$ rad, respectively. The output of the interferometer, gating signal and output of the DBM are shown in the upper, middle and lower traces of Fig. 5, respectively. Comparing Fig. 5 with Fig. 4 clearly reveals that the experimental data and the theoretical results closely correspond to each other.

For the conventional sinusoidal phase-modulated interferometric sensor with conventional LID technique, a band pass filter is deemed necessary to increase the signal to noise ratio. Since the central frequency of the band pass filter is sensitive to



(a)



(b)

Fig. 5. Experimental output of DBM under optical phase delay (a) $-\pi/2$ rad and (b) $\pi/2$ rad. The upper trace denotes the output of the interferometer, the middle trace represents the gating signal, and the lower trace is the output of the DBM.

temperature variation, the band pass filter was put in a temperature controller when the conventional LID technique was used in our Sagnac interferometer system. Figs. 6(a) and (b) summarize the results of the conventional LID technique and the PSDM, respectively. Fig. 6(a) displays the measured results when the optical phase delay of the Sagnac interferometer is maintained at 1.0×10^{-3} rad and temperature is stepwise changed from 20 to 65°C. This figure also reveals that the output is seriously drifting. In the PSDM, the output stability of the drift on the Sagnac interferometer system is upgraded to less than 1.0×10^{-5} rad. The temperature induced phase drift in Sagnac interferometer is thus significantly elevated.

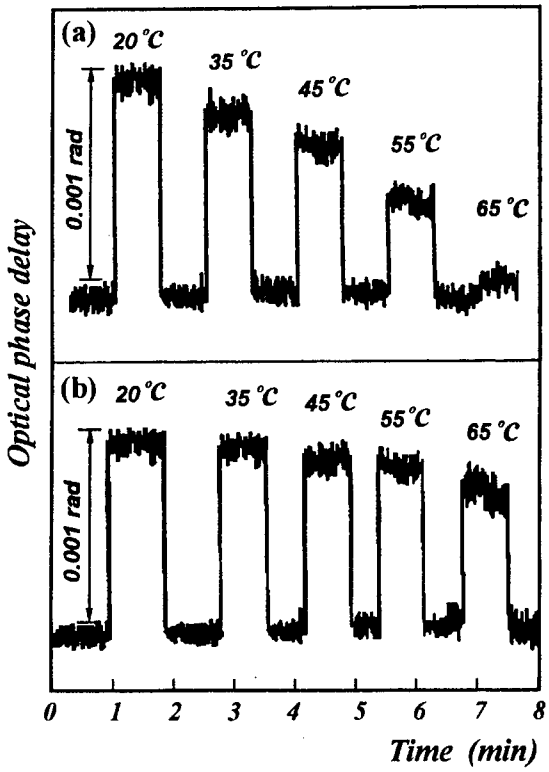


Fig. 6. Experimental results of (a) conventional phase sensitive detection and (b) phase sensitive detection method under optical phase delay 0.001 rad with various temperatures.

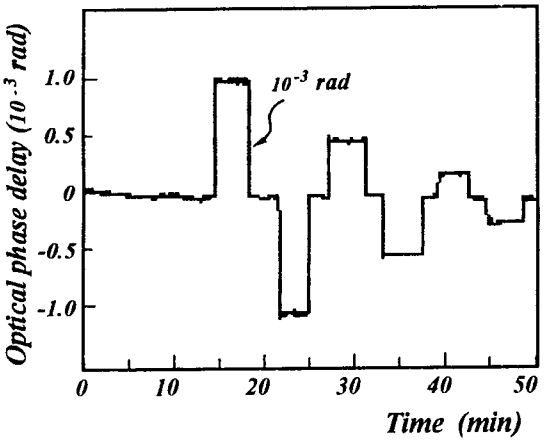


Fig. 7. Experimental output of Fig. 3 under various rotation rates.

Figure 7 presents the measured optical phase delays when the optical phase bias of the Sagnac interferometer system is stepwise changed at 1.0×10^{-3} rad, 0.5×10^{-3} rad, and 0.25×10^{-3} rad, respectively, under temperature 20°C. Fig. 8 compares the output characteristics with the measured output voltage V_o

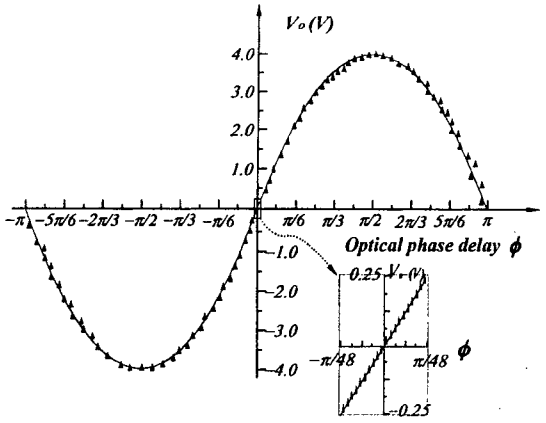


Fig. 8. Output characteristic between the output voltage and applied optical phase delay of Fig. 3.

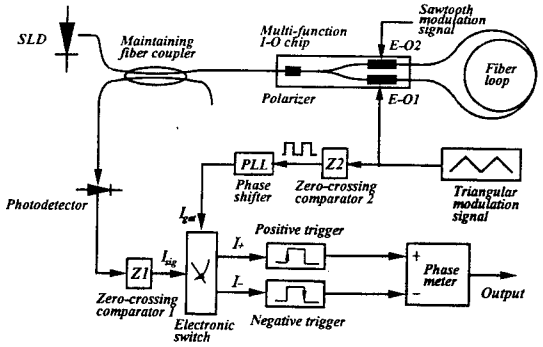


Fig. 9. Experimental setup of Sagnac interferometer system for direct phase difference measurement method. SLD: superluminescent diode, E-O: electrooptical phase modulator.

and applied optical phase delay ϕ of the PSDM. Obviously, the output characteristic is a sinusoidal function and linear variation is within the range of $\pm \pi/48$ rad. When the Sagnac interferometer is set at zero phase bias, the standard deviation of the low pass filter output is about 4.0×10^{-5} V/Hz$^{1/2}$; meanwhile, the mean value, with the applied optical phase change is selected at 0.25×10^{-3} rad, is 3.2×10^{-3} V/Hz$^{1/2}$. The sensitivity of the PSDM, which has the same order as that used in the conventional LID technique, is proximately 10^{-6} rad/Hz$^{1/2}$.

2. Direct phase difference method

Figure 9 illustrates the experimental setup for the direct phase difference method (DPDM). In this setup, the triangular modulation signal and constructed optical interferometric system are the same as those of the above mentioned PSDM. The detected optical interferometric signal using a photodetector is shaped into a square waveform using a zero-crossing comparator 1 (Z1). The output signal I_{sig} after Z1 is sent to the signal input portion of the electronic switch and, meanwhile, the triangular modulation signal is converted into a square signal using a zero-crossing comparator 2 (Z2). After a phase-locked loop (PLL) phase shifter, the gating signal I_{gat} has a $\pi/2$ rad phase shift compared with the triangular modulation signal. However, when the gating signal I_{gat} is used to control electronic switch, the output signal I_{sig} is gated into two separated signals I_+ and I_-. The rising and falling edges of each individual gating period are used as trigger sources to generate narrow pulse signals. Figs. 10(a), (b), (c) and (d) depict the output signals at photodetector and Z1, gating signal, outputs of the two gated signals and two pulse signals under an optical phase delay at $\phi=\pi/3$

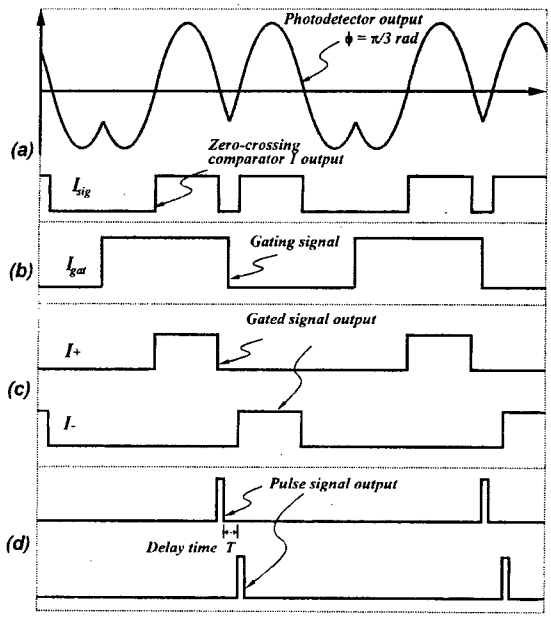


Fig. 10. Theoretical results of Fig. 9 under optical phase delay $\pi/3$ rad. (a) output signals of photodetector and zero-crossing comparator 1, (b) gating signal, (c) two output gated signals, and (d) two output pulse signals.

rad, respectively. Notably, properly selecting the pulse width of the output signal allows us to easily measure the optical phase delay difference using a phase meter. The output signal of the phase meter can be expressed as

$$V_o=K_S(2\phi+\phi_{offset}) \tag{6}$$

where K_S denotes the transfer gain of the phase meter, and ϕ represents the optical phase delay, and ϕ_{offset} is

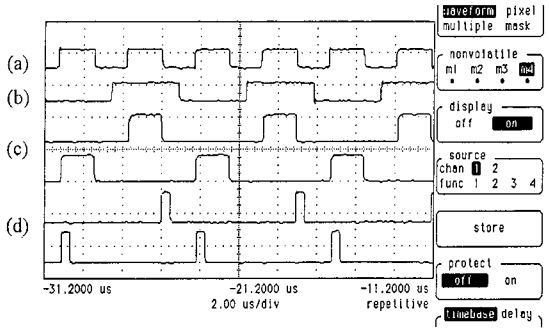


Fig. 11. Experimental results of Fig. 9 under zero optical phase delay. (a) output signal of zero-crossing comparator 1, (b) gating signal, (c) two output gated signals, and (d) two output pulse signals

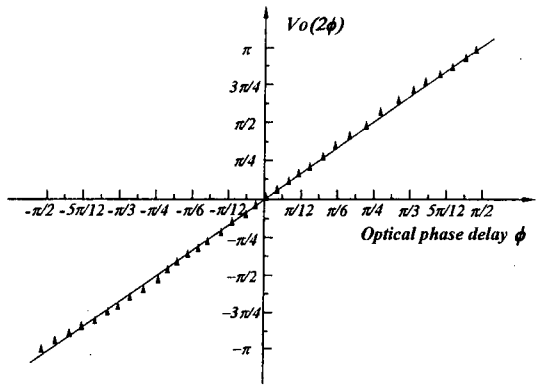


Fig. 12. Output characteristics between the output phase delay and applied optical phase delay of Fig. 9.

the optical phase offset generated in the optical interferometer. Moreover, the linear output characteristic can be obtained since ϕ_{offset} can be easily eliminated by an electronic subtracting circuit.

The characteristics and performance of DPDM can be summarized as follows:

(a) Sensitivity: According to Eq. (6), the sensitivity of the DPDM is limited by the resolution of the phase meter, which is about $\phi_{min} \cong 1.0\times10^{-5}\, rad/Hz^{1/2}$; it is one order lower than PSDM;

(b) Dynamic range: The maximum detectable optical phase delay in DPDM is $\phi_{max} \cong \pm\pi$ rad;

(c) The output curve of DPDM is linear; and

(d) Temperature dependence: Owing to no requirement of a band pass filter in the signal processing unit, the temperature dependent phase bias generated from the phase drift in the electronic circuit is the same as that of PSDM;

Figure 9 presents the experimental setup for DPDM, in which the pulse width of the two pulse signals is chosen as 1μ. Measuring the time delay difference between the two pulse signals allows us to derive the optical phase delay. Figs. 11(a), (b), (c) and (d) display the measured output signal of Z1, gating signal, two output gated signals, and two output pulse signals under zero optical phase delay, respectively. In the experiment, the various optical phase delay is obtained by altering the frequency of the sawtooth modulation signal employed at E-O2. By using a phase meter to measure the phase delay difference 2ϕ, its output characteristic between the output phase delay and applied optical phase delay is shown in Fig. 12. Obviously, the output of the phase meter is linear within the range of π rad.

3. Pseudo-heterodyne detection method

Figure 13 displays the pseudo-heterodyne

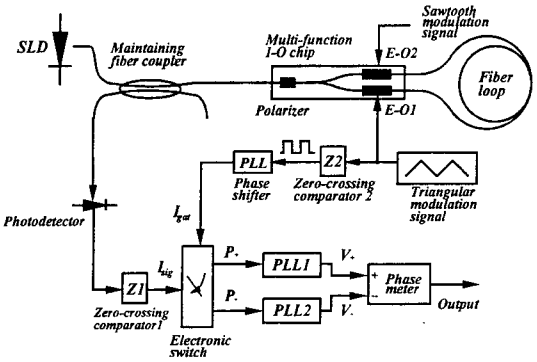


Fig. 13. Experimental setup of Sagnac interferometer system for pseudo-heterodyne detection method. SLD: superluminescent diode, E-O: electrooptical phase modulator.

detection method (PHDM). Its basic concept resembles that of DPDM except that the positive and negative triggers are replaced by phase-locked loop 1 (PLL1) and 2 (PLL2), respectively. The PLL1 and PLL2 are used as band pass filters. By using a photodetector to measure the output signal of the optical interferometric sensor, the measured signal is initially shaped into a square waveform signal I_{sig} using $Z1$ and then separated into two signals, P_+ and P_-, using an electronic switch controlled by the gating signal I_{gat}. The output signals, V_+ and V_-, are recovered when P_+ and P_- pass through PLL1 and PLL2, respectively. Figs. 14(a), (b), (c) and (d) illustrate the output signals of photodetector and $Z1$, gating signal, two output gated signals and two phase locked output signals under optical phase delay at $\phi=\pi/3$ rad, respectively. The output signals, V_+ and V_-, of the two PLL circuits can be expressed as

$$V_+ = K_P \cos(2\omega_m t + \phi),\ 0<t<T, \tag{7a}$$

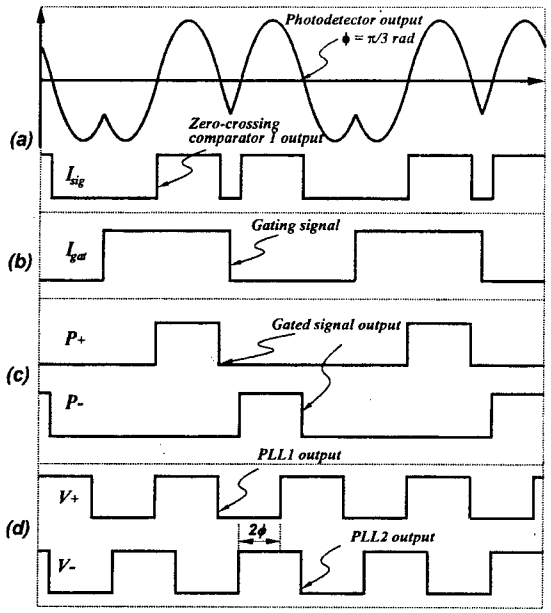


Fig. 14. Theoretical results of Fig. 13 under optical phase delay $\pi/3$ rad. (a) output signals of photodetector and zero-crossing comparator 1, (b) gating signal, (c) two output gated signals, and (d) two PLL output signals.

and

$$V_-=K_P\cos(2\omega_m t-\phi),\ 0<t<T, \tag{7b}$$

where K_P and ϕ denote the constant and the optical phase delay, respectively. The optical phase delay difference 2ϕ can be easily measured using a phase meter. If the modulation index is not set at 2π rad, the angular frequency of the two PLL outputs is shifted from $2\omega_m$ to $2\omega_m+\Delta\omega$, where $\Delta\omega<<2\omega_m$. Correspondingly, the output signals can be rewritten as

$$V_+=K_P\cos[(2\omega_m+\Delta\omega)t+\phi],\ 0<t<T, \tag{8a}$$

and

$$V_-=K_P\cos[(2\omega_m+\Delta\omega)\ t-\phi],\ 0<t<T. \tag{8b}$$

The nearest angular frequencies neighboring $2\omega_m+\Delta\omega$ are $\omega_m+\Delta\omega$ and $3\omega_m+\Delta\omega$, obviously accounting for why the difference between the two angular frequencies is still maintained as ω_m. When the locked-in range of the PLL exceeds $\Delta\omega$, the system can work well due to the phase lock of the output signal.

The characteristics and performance of PHDM can be summarized as follows:

(a) Sensitivity: According to Eq. (7) and Eq. (8), the sensitivity of the PHDM is the same as that of the DPDM;

(b) Dynamic range: The maximum detectable optical phase delay of the PHDM is $\phi_{max}=\pi$ rad, i.e. the same as that of the DPDM;

(c) The output curve of PHDM is linear;

(d) Temperature dependence: Owing to the lack of a requirement for a band pass filter in the signal processing unit, the temperature dependent phase bias generated from the phase drift in the electronic circuit is the same as that of PSDM; and

(e) From Eq. (8), when the modulation indexes of the triangular modulation signal are changed, the output signal is always tracked by the PLL technique, and is less sensitive to the change of modulation indexes. This observation also reveals that when the modulation index change is sustained at $\Delta(\alpha T/2)<0.1\pi$, then PHDM is still worked by the PLL tracking loop. However, for PSDM and DPDM, the modulation index should be sustained at $\Delta(\alpha T/2)<0.01\pi$ for the optical phase error less than 10^{-5} rad.

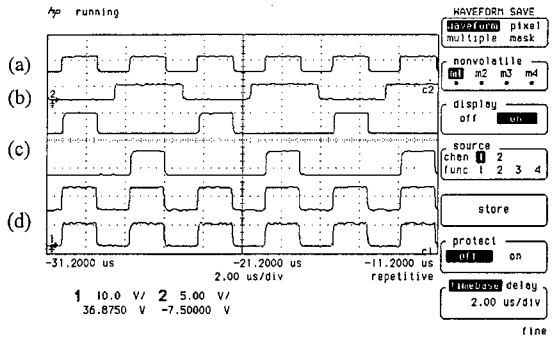


Fig. 15. Experimental results of Fig. 13 under zero optical phase delay. (a) output signal of zero-crossing comparator 1, (b) gating signal, (c) two output gated signals, and (d) two PLL output signals.

Figures 15(a), (b), (c) and (d) show the experimental output signal of Z1, gating signal, two output gated signals, and two PLL output signals under zero optical phase delay, respectively. By using a phase meter to measure the optical phase delay difference 2ϕ of the two PLL output signals, the output characteristics between the output phase delay and applied optical phase delay is shown in Fig. 16. This figure also reveals that the output of the phase meter is linear, i.e. similar to that observed in the PHDM, within the range of π rad.

Table 1 summarizes the overall performances of the three signal processing methods. According to this table, although the PSDM has the highest sensitivity, its dynamic range and linearity are poorer than DPDM and PHDM. Moreover, although DPDM

Table 1. Performances of PSDM, DPDM and PHDM

Item / Method	Sesitivity	Dynamic Range	Output Curve	Temperature Dependence	Modulation Index Dependence
PSDM	1.0 μrad	0.1 rad	Sinusoidal	Low (<5 μrad/°C)	High ($\Delta(\alpha T/2)<0.01\ \pi$)
DPDM	10 μrad	π rad	Linear	Low (<5 μrad/°C)	High ($\Delta(\alpha T/2)<0.01\ \pi$)
PHDM	10 μrad	π rad	Linear	Low (<5 μrad/°C)	Low ($\Delta(\alpha T/2)<0.1\ \pi$)

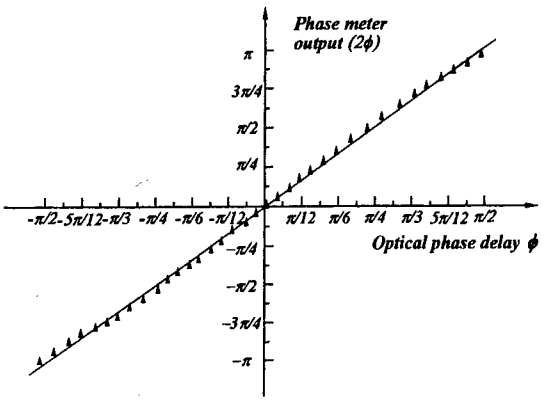


Fig. 16. Output characteristics between the output phase delay and applied optical phase delay of Fig. 13.

and PHDM yield the same performance in terms of sensitivity, linearity, and dynamic range, the latter is less sensitive to the phase modulation index variation of the triangular modulation signal.

III. CONCLUSIONS

This work presents three novel optical signal processes capable of detecting optical phase delay in an optical fiber interferometric sensor. These methods can also be applied to the other optical interferometric analyses in which the light source is frequency modulated. In the proposed methods, the triangular waveform signal, which is converted into a square gating signal, is used as a modulation signal to detect the optical phase delay. The optical phase delay can be measured by using the phase sensitive detection technique, dual pulse generation circuits followed by a time delay detection and the phase difference of carrier signals within a gated period. The methods proposed herein contain the following merits:

(1) The triangular modulation signal can be easily generated from an all digital electronic circuit;
(2) The output signal of the optical interferometer is a pure sinusoidal waveform;
(3) In the phase sensitive detection method and direct phase difference measurement method, the fact that the optical phase delay is directly measured from the output signal accounts for why it does not need a band pass filter which is sensitive to the environmental temperature; and
(4) In the pseudo-heterodyne detection method, because the carrier signal is phase locked, its angular frequency can be recovered to $2\omega_m$ even if the modulation index of the triangular modulation signal is not precisely located at 2π.

ACKNOWLEDGMENT

The authors would like to thank the National Science Council of the Republic of China for financially supporting this research under Contract No. NSC-84-2215-E008-006.

NOMENCLATURE

α	slope of the triangular modulation signal
T	period of the triangular modulation signal
I_o	optical power of light source
$I+$ & $I-$	output signal of the optical interferometric sensor
ϕ	optical phase delay
ν	frequency of light source
n	refractive index
L	optical path difference
ω_m	angular frequency of the modulation signal
f_c	characteristic frequency of fiber optic gyroscope
τ	propagation time in the fiber loop
$\phi(t)$ & $\phi(t-\tau)$	optical modulation phase of the counterclockwise and clockwise wave in the fiber loop
$\phi_{eff}(t)$	effective modulation phase
λ	wavelength of the light source
ΔI & I	stabilized and driving current of the light source

ΔT	stabilized temperature
V_o	averaged output signal
$\phi_{\min}$	minimum detectable optical phase delay
$\phi_{\max}$	maximum detectable optical phase delay
K_S	transfer gain of the phase meter
ϕ_{offest}	optical phase offset
V_+ & V_-	output signals of the two PLL circuits
K_p	transfer gain of the PLL circuit
$\Delta\omega$	angular frequency deviation
$\Delta(\alpha T/2)$	change of modulation index
I_{sig}	square signal output
P_+ & P_-	gated signal output
I_{gat}	gating signal

REFERENCES

1. Bergh, R.A., "Dual-ramp Closed-loop Fiber-optic gyroscope," Proc. SPIE, Vol. 1169, pp. 429-439 (1989).
2. Chien, P.Y. and C.L. Pan, "Multiplexed Fiberoptic Sensors using a Dual-slope Frequency-Modulated Source," Optics Letters, Vol. 16, No. 11, pp. 872-874 (1991).
3. Chien, P.Y., C.L. Pan and L.W. Chang, "Triangular Phase-modulated Approach to an Open-loop Fiberoptic Gyroscope," Optics Letters, Vol. 16, No. 21, pp. 1701-1703 (1991).
4. Chien, P.Y., Y.S. Chang and M.W. Chang, "Electrically Nulled Interferometric Sensor Based on Triangular Phase Modulation," Optics Communications, Vol.135, pp.198-202 (1997).
5. Dakin, J. and B. Culshaw, *Optical fiber sensors*, Vols 1 and 2 MA:Arteh House (1988).
6. Dandridge, A., A.B. Tveten, and T.G. Giallorenzi, "Homodyne Demodulation Scheme for Fiber Optic Sensors using Phase Generated Carrier," *IEEE Journal of Quantum Electronics*, Vol. 18, No. 10, pp. 1647-1653 (1982).
7. Davis, M.A. and A.D. Kersey, "Application of a Fourier Transform Spectrometer to the Detection of Wavelength-encoded Signals from Bragg Grating Sensors," *Journal of Lightwave Technology*, Vol. 13, No. 7, pp. 1289-1295 (1995).
8. Ezekiel, S. and H.J. Arditty, "Fiber-optic Rotation Sensors," *Springer-Verlay*, pp. 7-12 (1982).
9. Giallorenzi, T.G., J.A. Bucaro, A. Dandridge, G.H. Sigel, JR., J.H. Cole, S.C. Rashleigh, and R.G. Priest, "Optical Fiber Sensor Technology," *IEEE Journal of Quantum Electronics*, Vol. 18, No. 4, pp. 626-665 (1982).
10. Jackson, D.A., "Recent Progress in Monomode Fiberoptic Sensors," *Measurement Science and Technology*,--- Vol. 5, No. 6, pp. 621-638 (1994).
11. Jackson, D.A., R. Priest. A. Dandridge and A.B. Tveten, "Elimination of Drift in a Single-mode Optical Fiber Interferometer using a Piezoelectrically Stretched Coiled Fiber," *Applied Optics*, Vol. 19, No. 17, pp. 2926-2929 (1980).
12. Jin, W. and B. Culshaw, "Frequency-division Multiplexing of Fiberoptic Gyroscopes," *Journal of Lightwave Technology*, Vol. 10, No. 10, pp. 1473-1480 (1992).
13. Jin, W., D. Uttamchandani, and B. Culshaw, "Direct Readout of Dynamic Phase-changes in a Fiberoptic Homodyne Interferometer," *Applied Optics*, Vol. 31, No. 34, pp. 7253-7258 (1992).
14. Kersey, A.D., A. Dandrige and A.B. Tveten, "Overview of Multiplexing Techniques for Interferometric Fiber Sensors," Proc. SPIE, Vol. 838, pp. 184-193 (1987).
15. Kersey, A.D., D.A. Jackson and M. Corke, "Demodulation Scheme Fiber Interferometric Sensors Employing Laser Frequency Switching," *Electronics* Letters, Vol. 19, No. 3, pp. 102-103 (1983).
16. Kim, B.Y. and H.J. Shaw, "Phase-reading All-fiber-optic Gyroscope," Optics Letters, Vol. 9, No. 8, pp. 378-380 (1984).
17. Lefe'vre, H.C., P. Martin, J. Morisse, P. Simonpie'tri, P. Vivenot, and H. J. Arditty, "High Dynamic Range Fiber Gyro with all-digital Signal Processing," Proc. SPIE, Vol. 1367, pp. 72-80 (1990).
18. Santos, J.L., F. Farahi, T. Newson, A.P. Leite, and D.A. Jackson, "Frequency Multiplexing of Remote all-fiber Michelson Interferometers with Lead Insensitivity," *Journal of Lightwave Technology*, Vol. 10, No. 6, pp. 853-863 (1992).
19. Stubbe, R., G. Edwall, B. Sahlgren, and L. Svahn, "A Pseudo-Heterodyne Fiber Optical Gyro using Integrated-optics," *Journal of Lightwave Technology*, Vol. 10, No. 10, pp. 1489-1498 (1992).
20. Uttam, D. and B. Culshaw, "Precision Time Domain Reflectometry in Optical Fiber System using a Frequency Modulated Continuous wave Ranging Technique," *Journal of Lightwave Technology*, Vol. 3, No. 5, pp. 971-978 (1985).
21. Wei, J., M.Z. Li, D. Uttamchandani, and B. Culshaw, "Modified J1...J4 Method for Linear Readout of Dynamic Phase-changes in a Fiberoptic Homodyne Interferometer," Applied Optics, Vol. 30, No. 31, pp. 4496-4499 (1991).
22. Weis, R.S., A.D. Kersey and T.A. Berkoff, "A four-element Fiber Grating Sensor Array with Phase-sensitive Detection," *IEEE Photonics Technology Letters*, Vol. 6, No. 12, pp. 1469-1472 (1994).

Discussions of this paper may appear in the discussion section of a future issue. All discussions should be submitted to the Editor-in-Chief.

Manuscript Received: June 25, 1997
Revision Received: Dec. 09,1997
and Accepted: Jan. 24, 1998

三角波相位調制技術應用於干涉式光纖感測器

李清庭　張立武

國立中央大學光電所

簡碧堯

中山科學研究院材料研發中心

摘　要

本文針對光學干涉式感測器中相位差的測量，提出三種新穎的信號處理檢測技術。在本系統裡，使用三角波信號來調制光相位及作爲光相位差的量測。在文章裡第一種技術係針對干涉儀的輸出信號作分割，並經過一低通濾波器以取得光相位差，其次是測量分割信號間的時間差以量得光相位差，最後是使用超外差的方法測量輸出信號間的相位差以得到光相位差。在整個實驗過程中，使用較少受外界環境影響的光纖Sagnac干涉式感測器進行實驗的量測。由實驗的結果，證實本系統的信號處理技術可得到很高的靈敏度和良好的線性度。

關鍵詞：超光二極體、光纖陀螺儀、相位鎖入迴路、積光相位調制器。

REACTION OF CARBON DISULFIDE AND O-PHENYLENE DIAMINE BY TERTIARY AMINE IN THE PRESENCE OF POTASSIUM HYDROXIDE

Biing-Lang Liu and Maw-Ling Wang*
Department of Chemical Engineering
National Tsing Hua University
Hsinchu, Taiwan 300, R.O.C.

Key Words: Synthesis of 2-mercaptobenzimidazole (MBI), tertiary amines, potassium hydroxide.

ABSTRACT

The reaction of carbon disulfide and o-phenylene diamine catalyzed by tertiary amine in the presence of KOH in an aqueous solution/organic solvent two-phase medium was carried out. The reaction was greatly enhanced by adding a small amount of tertiary amine in the presence of KOH. The reaction of synthesizing 2-mercaptobenzimidazole (MBI) first took place in the organic phase. However, the potassium salt of MBI, which was produced from the reaction of MBI and KOH at the interface between CH_2Cl_2 and H_2O, dissolved in the aqueous phase. The greatest advantage of using this process is that MBI in crystal form can then be precipitated from the aqueous solution by adding an acidic compound. Based on the experimental data, a reaction mechanism was proposed. The reaction of synthesizing MBI was first initiated by reacting CS_2 and R_3N to produce an active intermediate (R_3N-CS_2). This active intermediate further reacted with o-phenylene diamine to produce the desired MBI product. In addition, potassium hydroxide also reacted with H_2S, which is a byproduct from the synthesis of MBI, to enhance the reaction. The reaction of CS_2 and $C_6H_4(NH_2)_2$ in a two-phase medium is described by a pseudo-first-order rate law.

I. INTRODUCTION

Quaternary ammonium salts are widely used to synthesize specialty chemicals from two immiscible reactants [4, 13, 14, 17]. A steadily increasing number of papers related to the applications of quaternary ammonium salts have been published in recent literature and documents [5, 7, 8]. However, these processes usually suffer some of the same disadvantages as the conventional homogeneous and heterogeneous catalysts, mainly separation of the product from the solution. Advanced techniques of chemical separation are required to purify the product.

2-Mercaptobenzimidazole (MBI) is an important specialty chemical which is extensively used in industry as an inhibitor, antioxidant, antiseptic and adsorbent [10, 11, 15, 18]. Synthesis of MBI and its derivatives has been reported in recent years. The procedure involved is, namely, the reaction of

*Correspondence addressee

o-phenylene diamine and the reactant in a cosolvent of CH_3OH and H_2O catalyzed by quaternary ammonium salts or active carbon (Norit) [16]. However, this reaction was carried out at a high temperature for a long time period. Later, the synthesis of MBI was carried out from the reaction of o-phenylene diamine and carbon disulfide in an alkaline solution [12], and employed quaternary ammonium hydroxide as the catalyst [6]. Related reactions synthesizing the derivatives of MBI have also been reported [1, 2, 19]. However, the generally accepted reaction mechanism is not available.

In our preliminary work toward the synthesis of MBI, it was found that the reaction of o-phenylene diamine and carbon disulfide, was catalyzed by a tertiary amine. Furthermore, the rate of the reaction was greatly enhanced by adding KOH in an appropriate amount. The primary objective of this work was to improve the synthesis process for producing MBI by reacting o-phenylene diamine and carbon disulfide catalyzed by tertiary amine in the presence of KOH in a two-phase medium. The potassium salt of MBI dissolved in the aqueous phase. The greatest advantage of using this process is that MBI in crystal form then precipitated from the aqueous solution after adding an acidic compound. It can be easily separated by mechanical filtration and centrifuge. The key point was to investigate the reaction mechanism and kinetics of the reaction of o-phenylene diamine and carbon disulfide catalyzed by tertiary amine in the presence of potassium hydroxide. The effects of the operating conditions on the reaction are examined and the optimum conditions to obtain a high product yield are discussed.

II. EXPERIMENTAL SECTION

1. Materials

Carbon disufide (CS_2), o-phenylene diamine ($C_6H_4(NH_2)_2$), potassium hydroxide, tertiary amines including TEA ($(C_2H_5)_3N$), TPA ($(C_3H_7)_3N$) and TBA ($(C_4H_9)_3N$) and other reagents are all G.R. grade chemicals for synthesis.

2. Procedures

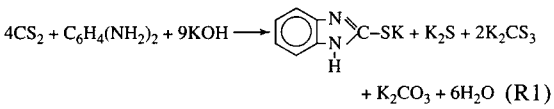
The reactor was a 125 mL four necked Pyrex flask able to serve the purposes of agitating the solution, inserting the thermometer, taking samples, and feeding the reactants. A reflux condenser was attached to the port of the reactor to recover carbon disulfide. The reactor was submerged into a constant temperature water bath in which the temperature could be controlled to ±0.1°C. To start an experimental run, known quantities of o-phenylene diamine, carbon disulfide, caffeine (external standard method), and tertiary amine were dissolved in organic solvent and introduced into the reactor. The mixture was stirred mechanically by a two-blade paddle (5.5 cm) at 1000 rpm. During the reaction, an aliquot sample of 0.1 mL was withdrawn from the solution at a chosen time. The sample was immediately introduced into a methanol solvent at 4°C for dilution and to retard the reaction, then the sample was analyzed by HPLC.

The product 2-mercaptobenzimidazole (MBI) for identification was purified from the reaction solution without containing tertiary amine by vacuum evaporation to strip off organic solvent and carbon disulfide. Then, it was dissolved into ethanol prepared for recrystallization. A white crystal form of MBI, which was insoluble in EtOH, was obtained by cooling the solution.

The product (salt of MBI) and reactants (CS_2 and $C_6H_4(NH_2)_2$) were identified by NMR and IR and the contents of the reactants and product were analyzed by an HPLC instrument. The results obtained from NMR and IR are very consistent with the published data. An HPLC model LC9A (Shimadzu) with an absorbance detector (254 nm, SPD-6A) was employed to measure the contents of reactants and product. The column used was Shim-pack LCL-ODS RP-18 (5 μm). The eluent was CH_3CN/H_2O=20/80 (with 5 mM KH_2PO_4+0.1% H_3PO_4) (volume ratio) with a flow rate 1.0 mL/min.

III. KINETICS MODEL OF REACTION

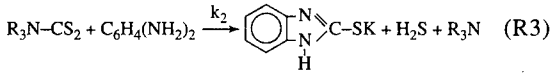
The overall reaction is expressed as



$$4CS_2 + C_6H_4(NH_2)_2 + 9KOH \longrightarrow \text{(MBI-SK)} + K_2S + 2K_2CS_3 + K_2CO_3 + 6H_2O \quad (R1)$$

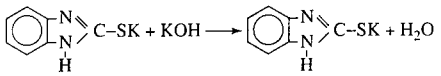
In the organic-phase reaction synthesizing MBI, it is assumed that carbon disulfide first reacts with the tertiary amine to form an active intermediate (R_3N–CS_2). This active intermediate further reacts with o-phenylene diamine to produce the desired product MBI, i.e., [9]

$$R_3N + CS_2 \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} R_3N\text{–}CS_2 \quad (R2)$$



$$R_3N\text{–}CS_2 + C_6H_4(NH_2)_2 \xrightarrow{k_2} \text{(MBI-SK)} + H_2S + R_3N \quad (R3)$$

The reaction of MBI and KOH to produce a water-soluble potassium salt of MBI takes place on the organic-aqueous interface, i.e.



(R4)

$$2KOH+H_2S \longrightarrow K_2S+2H_2O \quad (R5)$$

$$3CS_2+6KOH \longrightarrow 2K_2CS_3+K_2CO_3+3H_2O \quad (R6)$$

Reactions (R2) and (R3) are the two main steps of MBI synthesis in the organic solution. Reactions (R5) and (R6) are the two side reactions which also take place on the organic-aqueous interface and in the aqueous phase. In general, the reaction (R2) is very fast and reaches an equilibrium state within 3-4 minutes. Usually, it took about 6 h to obtain a 80% conversion of o-phenylene diamine at moderate reaction conditions [9]. Therefore, we proposed that the reaction of carbon disulfide and tertiary amine (reaction (R2)) is in an equilibrium state relative to reaction (R3). Reaction (R4) was carried out to obtain the potassium salt of MBI. The reaction (R3) was enhanced by reacting KOH and H_2S which is given in reaction (R5). In reaction (R6), inert compounds K_2CS_3 and K_2CO_3 which did not possessing a catalytic properties were produced from the reaction of CS_2 and KOH.

The rates of reactions (R2) and (R3) are expressed as

$$r_1=k_1[CS_2]\,[R_3N]-k_{-1}[R_3N-CS_2] \quad (1)$$

$$r_2=-\frac{d[C_6H_4(NH_2)_2]}{dt}=k_2[R_3N-CS_2][C_6H_4(NH_2)]] \quad (2)$$

Assume that a pseudo-steady-state hypothesis is applied for the active intermediate R_3N-CS_2, i.e.,

$$\frac{d[R_3N-CS_2]}{dt}=0. \quad (3)$$

No other byproducts were obtained. Therefore, it is reasonable to assume that the consumption rate of R_3N-CS_2 in (R_3) equals the production rate of R_3N–CS_2 in (R2), i.e.,

$$k_1[R_3N][CS_2]-k_{-1}[R_3N-CS_2]$$

$$=k_2[R_3N-CS_2]\,[C_6H_4(NH_2)_2]. \quad (4)$$

Rearranging Eq. (4), we obtain

$$r=r_1=r_2=\frac{k_2[C_6H_4(NH_2)_2]\,[R_3N]\,[CS_2]}{(k_2/k_1)\,[C_6H_4(NH_2)_2]+(k_{-1}/k_1)}. \quad (5)$$

Reaction (R3) is very slow relative to reaction (R2). It is obvious that reaction (R3) is a rate-determining step in the organic-phase reaction. Thus, it is reasonable to assume that $k_{-1}[R_3N-CS_2]>>k_2[R_3N-CS_2]$ $[C_6H_4(NH_2)_2]$ or

$$(k_{-1}/k_1)=K>>(k_2/k_1)[C_6H_4(NH_2)_2]. \quad (6)$$

Thus, Eq. (5) is rewritten as

$$r=(k_2/K)[R_3N][CS_2][C_6H_4(NH_2)_2]. \quad (7)$$

In this work, carbon disulfide was usually used in a large excess amount relative to its stoichiometric quantity. Also, the concentration of R_3N remained constant. Therefore, Eq. (7) is written as

$$r=-\frac{d[C_6H_4(NH_2)_2]}{dt}=k_{app}[C_6H_4(NH_2)_2] \quad (8)$$

where

$$k_{app}=(k_2/K)[R_3N][CS_2]. \quad (9)$$

Integrating Eq. (8), the reaction is expressed by a pseudo-first-order rate law,

$$-1n(1-X)=k_{app}t \quad (10)$$

where X is the conversion of o-phenylene diamine and is defined as

$$X=1-([C_6H_4(NH_2)_2]/[C_6H_4(NH_2)_2]_i) \quad (11)$$

in which the subscript "i" denotes the initial condition. A plot of $-1n(1-X)$ vs. time leads to a straight line with slope k_{app}. Experimental results indicated that the same results were obtained for the production rate determined from the product and for the consumption rate determined from the reactant. We confirmed that no other byproducts were produced during or after the reaction.

IV. RESULTS AND DISCUSSION

1. Identification of the reaction mechanism

In the reaction of CS_2 and $C_6H_4(NH_2)_2$ to synthesize MBI, the reaction rate was slow in the absence of an alkaline compound and catalyst (tertiary amine). Only a 10% conversion of o-phenylene diamine was obtained after 8 hours of reaction. As shown in Table 1, the reaction was enhanced by adding a small amount of tributylamine (Bu_3N).

As previously stated, the mechanism of the reaction of CS_2 and $C_6H_4(NH_2)_2$ by a tertiary amine was described by reactions (R2) and (R3). We assume that the reaction of R_3N and CS_2 is fast and reaches an equilibrium state within 3-4 minutes or more. The active intermediate R_3N-CS_2 cannot be isolated and

Table 1. Effects of the operating parameters on the k_{app}-values; 0.4 g of $C_6H_4(NH_2)_2$, 4.003 g of CS_2, 0.8188 g of KOH, 50 mL of CH_2Cl_2, 20 mL of H_2O, 1000 rpm, 30°C (Except for the effect of a particular factor on the conversion, the other conditions are specific in the above.)

TBA (g)	0	0.156	0.311	0.467	0.622	0.778	0.934
kapp (1/hr)	0.104* 0.012**	0.1278	0.1361	0.1639	0.1875	0.2111	0.2333
CS_2, g		0.4068	1.8121	2.4003	3.0064	4.0029	5.0029
kapp (1/hr)		0.0093	0.0743	0.0947	0.1173	0.1373	0.1813
$C_6H_4(NH_2)_2$, g		0.2	0.4	0.6	0.75	0.9	1.0
kapp (1/hr)		0.1042	0.1375	0.1861	0.2069	0.2251	0.2319
Solvents		CH_2Cl_2	$CHCl_3$	C_6H_6	C_6H_5Cl	$C_6H_5CH_3$	
kapp		0.1365	0.0973	0.1365	0.1676	0.1622	

*: without TBA
**: without TBA and KOH

purified from the solution. However, the presence of an orange color, indicated that the R_3N-CS_2 product appeared in the solution. The UV-absorbance of CS_2 and R_3N solution was increased during the reaction when the mixed solution was scanned by a UV instrument. The UV-absorbance indicates that the reaction reached an equilibrium state within a short time period of about 5 minutes.

To further examine the equilibrium reaction (R2), two experimental runs in different sequential orders of reaction procedures were tested in this study. The first one involved conducting the reaction of carbon disulfide, tributylamine (Bu_3N) and potassium hydroxide for 2 hrs in advance, and the o-phenylene diamine was added to the reaction solution. The second procedure involved the simultaneous introduction of CS_2, Bu_3N, KOH and $C_6H_4(NH_2)_2$ dissolved in CH_2Cl_2/H_2O (v/v=50/20) to start the reaction. Both experimental runs resulted in the conversion of o-phenylene diamine, and there were no differences in the conversion. These results indicate that the reaction of CS_2 and R_3N to produce the active intermediate (R_3N–CS_2) is very fast. The reaction of the active intermediate (R_3N–CS_2) and o-phenylene diamine is a rate-determining step for the entire reaction. KOH only acts as the neutralizing agent is the reaction with H_2S, which is a byproduct of the reaction of CS_2 and $C_6H_4(NH_2)_2$. The reaction of KOH and H_2S to enhance the synthesis of MBI is attributed to LeChatelier's principle. Therefore, two fundamental reaction steps were proposed to represent the entire reaction given by (R2) and (R3).

The distribution of $C_6H_4(NH_2)_2$ between the CH_2Cl_2 and H_2O two-phase solution was examined [18]. It was found that $C_6H_4(NH_2)_2$ dissolves both in CH_2Cl_2 and H_2O. It takes only 30 seconds for $C_6H_4(NH_2)_2$ to reach an equilibrium state between two phases. The resistance of mass transfer of o-phenylene diamine between two phases is negligible. Both CS_2 and R_3N–CS_2 do not dissolve in water. Therefore, the reaction of R_3N–CS_2 and $C_6H_4(NH_2)_2$ takes place in the CH_2Cl_2 phase rather than in the aqueous phase. o-Phenylene diamine, which dissolves in the aqueous solution was counted as a source in providing the reactant in the organic phase.

2. Factors affecting the reaction kinetics in synthesizing MBI

(i) Identification of the reaction mechanism

As stated, the potassium salt of MBI dissolved in the aqueous phase. It is interesting to investigate the reaction mechanism of the reaction of o-phenylene diamine and carbon disulfide catalyzed by the tertiary amine in the presence of the KOH solution/CH_2Cl_2 two-phase medium. Our preliminary study [9] showed that the tertiary amine first reacted with carbon disulfide to form an active intermediate (R_3N-CS_2), which is dissolved in the organic solvent rather than in the aqueous phase.

In this study, an experiment was conducted, i.e., R_3N in a limiting amount dissolved in CS_2 in a large excess to form a homogeneous solution. For this a solution containing CS_2 and R_3N was prepared. After stripping off CS_2 from the organic solution, the residue solution was poured into an aqueous solution containing o-phenylene diamine. Only a trace amount of MBI was produced. The reason is that CS_2 is stripped to a limiting value. The production of MBI in the aqueous phase is limited even if the tertiary

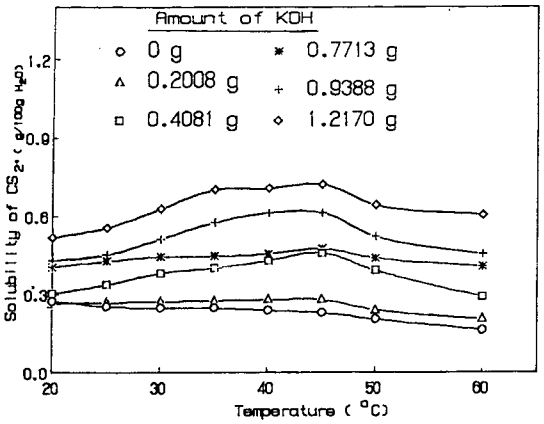


Fig. 1. Effect of the amount of potassium hydroxide on the solubility of carbon disulfide in the aqueous solution; 6.3 g of CS_2, 100 mL of water

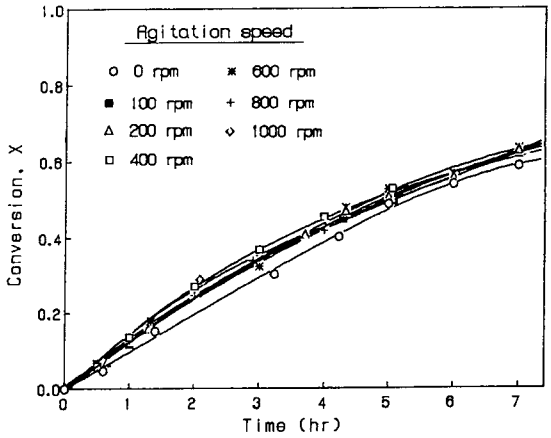


Fig. 2. Effect of the agitation speed on the conversion of o-phenylene diamine; 0.4 g of $C_6H_4(NH_2)_2$, 0.311 g of TBA, 0.8188 g of KOH, 50 mL of CH_2Cl_2, 20 mL of H_2O, 30°C

amine (or R_3N-CS_2) is present in the solution. In the other experiment, the organic solution of CS_2 containing R_3N-CS_2 was poured into the aqueous solution containing o-phenylene diamine. A relatively large amount of MBI was produced from the aqueous solution. This experiment indicated that the reaction of o-phenylene diamine and carbon disulfide catalyzed by the tertiary amine took place in the aqueous phase or at the organic-aqueous interface. Nevertheless, it is obvious that the reaction rate in the aqueous phase and the yield of MBI are highly dependent on the the amount of CS_2 in the aqueous phase. Therefore, we confirmed that the organic phase reaction of CS_2 and $C_6H_4(NH_2)_2$ is a rate-determining step in the two-phase medium because most CS_2 stays in the organic solution. The yield of MBI produced from the aqueous phase is low. Fig. 1 shows the solubility of CS_2 in the alkaline solution of KOH. Little of the CS_2 dissolved in the aqueous solution. Furthermore, the dissolved CS_2 in the aqueous phase also reacts with KOH to form K_2CS_3 and K_2CO_3 [3]. Therefore, part of the CS_2, which dissolved in the aqueous phase, is consumed. We conclude that the yield of MBI produced from the aqueous phase is low. The yield of MBI in the two-phase reaction comes mostly from the organic-phase reaction.

As stated, the yield of MBI product from the aqueous phase is low. Our preliminary study for the reaction of o-phenylene diamine and carbon disulfide catalyzed by tertiary amine in a homogeneous solution containing ($MeOH+H_2O$) found that the reaction rate is decreased with the increase in the amount of water in the homogeneous solution [9]. This result indicates that the reaction rate is still low in the aqueous phase. Therefore, the production of MBI comes directly from the organic phase.

(ii) Effect of agitation speed

Figure 2 shows the effect of agitation on the conversion of o-phenylene diamine. Not much difference in the conversion was obtained when the agitation speed changed from 100 to 1000 rpm. This result also implies that the reaction of o-phenylene diamine and CS_2 in the organic phase plays an important role. The ionic reaction of KOH and H_2S or the reaction of KOH and MBI on the organic-aqueous interface is very fast even at low agitation speed. The reaction of the rate-determining step took place in the organic phase.

(iii) Effect of the amount of tributylamine

Table 1 shows the effects of tributylamine (TBA) in the presence of KOH on the conversion of o-phenylene diamine in the two-phase reaction medium. The reaction follows a pseudo-first-order rate law. The conversion is increased with the increase in the amount of tributylamine. It is obvious that the tertiary amine is an effective catalyst in enhancing the reaction rate. The basic tertiary amine reacted with CS_2 to produce an active intermediate R_3N–CS_2. Table 1 shows the effects of the operating parameters, such as: amount of Bu_3N, content of CS_2 and $C_6H_4(NH_2)_2$, and organic solvents, on the value of k_{app}.

(iv) Effect of the amount of potassium hydroxide

Figure 3 shows the effect of the amount of KOH on the conversion of o-phenylene diamine in the presence of tertiary amine at 4, 5, 6, 7, hrs of reaction.

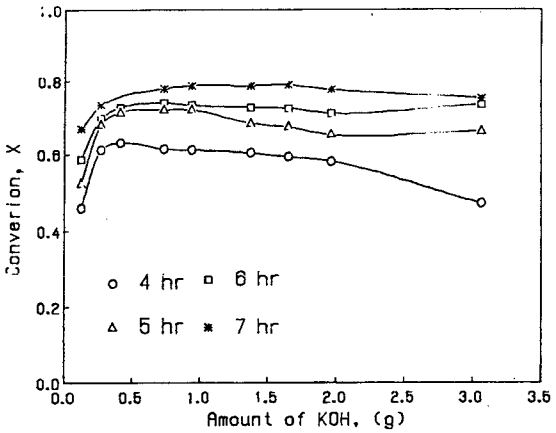


Fig. 3. Effect of the amount of KOH on the conversion of o-phenylene diamine; 0.4 g of $C_6H_4(NH_2)_2$, 0.311 g of TBA, 6.3 g of CS_2, 50 mL of CH_2Cl_2, 20 mL of H_2O, 1000 rpm, 30°C

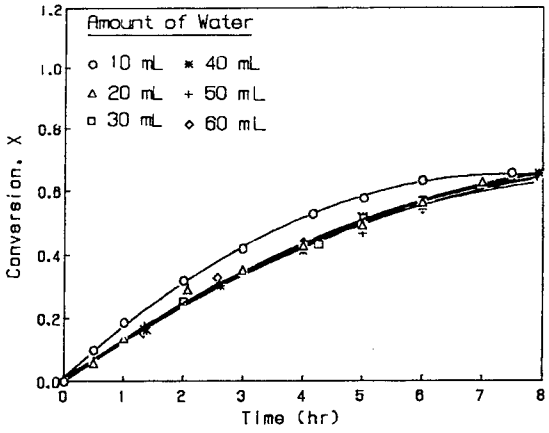


Fig. 4. Effect of the amount of water on the conversion of o-phenylene diamine; 0.4 g of $C_6H_4(NH_2)_2$, 0.311 g of TBA, 4.003 g of CS_2, 0.8188 g of KOH, 50 mL of CH_2Cl_2, 20 mL of H_2O, 1000 rpm, 30°C

The conversion is relatively low when no KOH or only a small amount of KOH is added. However, reactions (R4) and (R5) show that the reaction is enhanced by adding an appropriate amount of KOH following LeChatelier's principle. The conversion is then decreased when a larger amount of KOH is added, i.e., the number of moles of KOH is larger than that of CS_2. Potassium hydroxide reacted with carbon disulfide to produce K_2CS_3 and K_2CO_3. Parts of CS_2 are therefore consumed by reacting with KOH. In this study, the moles of KOH were less than that of CS_2. The reaction synthesizing the MBI product is not completely retarded by adding KOH. An appropriate amount of KOH leads to a relatively high yield of MBI product.

(v) Effect of the amount of carbon disulfide and o-phenylene diamine

In the two-phase reaction, the effects of the concentration of CS_2 on the conversion is shown in Table 1. The conversion of o-phenylene diamine is increased with the increase in the concentration of CS_2. In deriving the kinetic model, the rate expression is depicted in Eq. (5), or in Eqs. (9) and (10). It is shown that the reaction rate is directly proportional to the concentration of CS_2. This is consistent with the experimental data.

The effect of the concentration of o-phenylene diamine on the conversion of o-phenylene diamine is also shown in Table 1. At 30°C, the conversion is increased with an increase in the concentration of o-phenylene diamine. We expected that this tendency would be the same at higher temperatures, i.e., the conversion would increase with the increase in the concentration of o-phenylene diamine for the whole temperature range. Similarly, the experimental data follow pseudo-first order rate law. A plot of $-\ln(1-X)$ vs. time of reaction shows a straight line of slope k_{app} which is depicted in Table 1.

(vi) Effect of the amount of water

Fig. 4 shows the effect of the amount of water on the conversion of o-phenylene diamine. In general, the conversion of o-phenylene diamine is decreased with an increase in the amount of water. The general accepted reason is that o-phenylene diamine dissolves both in aqueous and organic solution. However, the reaction only takes place in the organic phase. It is obvious that the amount of o-phenylene diamine, which exists both in the aqueous phase, acts as the source in providing the reactant in the organic phase. Therefore, the number of moles of o-phenylene diamine is increased in the organic phase when the amount of water is decreased. In addition, the concentration of KOH in the aqueous phase is also increased to enhance the reaction rate in using a small amount of water. For this, the conversion of o-phenylene diamine is increased with the decrease in the amount of water. However, this effect is not signification.

(vii) Effects of the organic solvents

In this work, chlorobenzene, dichloromethane, toluene, chloroform and benzene served as the organic solvents. Table 1 shows the effects of the orgainc solvents on the conversion of o-phenylene diamine. The order of the reactivity for the reaction in these

Table 2. Effect of temperatures on the k_{app}-values for the reactions in various organic solvents; 0.4 g of $C_6H_4(NH_2)_2$; 4.003 g of CS_2, 0.8188 g of KOH, 0.311 g of TBA, 50 mL of organic solvent, 20 mL of H_2O, 1000 rpm

	k_{app}-values (1/hr)			
Solvents	30°C	35°C	40°C	45°C
Chlorobenzene	0.1689	0.2122	0.2784	0.3486
Chloroform	0.0973	0.1392	0.1649	0.1946
Benzene	0.1378	0.1743	0.2378	0.2946
Toluene	0.1622	0.2176	0.2716	0.3378
TEA*	0.25	0.3014	0.3554	0.4540
TPA*	0.1648	0.2040	0.2581	0.3149
TBA*	0.1689	0.2122	0.2784	0.3486

* 1.68×10^{-3} mole of tertiary amine, organic solvent: chlorobenzene

organic solvents is: chlorobenzene > toluene > benzene > dichloromethane > chloroform. The order of the dielectric constants for these organic solvents is: dichloromethane > chloroform > benzene > chlorobenzene > toluene. The experimental results indicate that the rate of reaction does not correspond to the dielectric constant of the organic solvent.

(viii) Effect of temperature

The effect of temperature on the conersion of o-phenylene diamine for the reaction using chlorobenzene as the organic solvent is shown in Table 2. The increase in temperature enhances the reaction rate and the conversion of o-phenylene diamine. The reaction also follows pseudo-first-order rate law. Similar results were obtained using other organic solvents such as chloroform, benzene and toluene, and using catalysts such as TEA, TPA and TBA. Table 2 shows the k_{app}-value at various temperatures for the reaction carried out in different orgainc solvents. An Arrhenius plot of $-\ln(k_{app})$ vs. 1/T at various organic solvents and catalysts is shown in Figs. 5 and 6. The Arrhenius equation for various organic solvents and for various tertiary amines are:

Organic solvents:

Chlorobenzene: $k_{app}=9.77\times10^5\exp(-4.72\times10^3/T)$

Chloroform: $k_{app}=1.83\times10^5\exp(-4.36\times10^3/T)$

Benzene: $k_{app}=2.00\times10^6\exp(-5.00\times10^3/T)$

Toluene: $k_{app}=8.46\times10^5\exp(-4.68\times10^3/T)$

The activation energies in using those organic solvents are 35.76 ~ 41.80 KJ/mole.

Tertiary amines:

TEA: $k_{app}=6.89\times10^4\exp(-3.80\times10^3/T)$

TPA: $k_{app}=1.73\times10^5\exp(-4.20\times10^3/T)$

TBA: $k_{app}=9.77\times10^5\exp(-4.72\times10^3/T)$

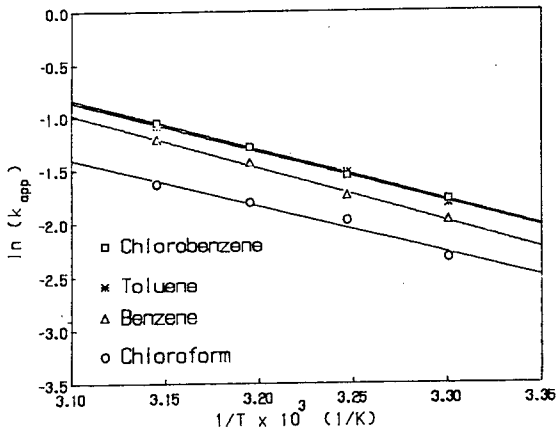


Fig. 5. Arrhenius plot of $\ln(k_{app})$ vs. 1/T for the reaction in various organic solvents; 0.4 g of $C_6H_4(NH_2)_2$, 4.003 g of CS_2, 0.311 g of TBA, 0.8188 g of KOH, 50 mL of organic solvent, 20 mL of H_2O, 1000 rpm.

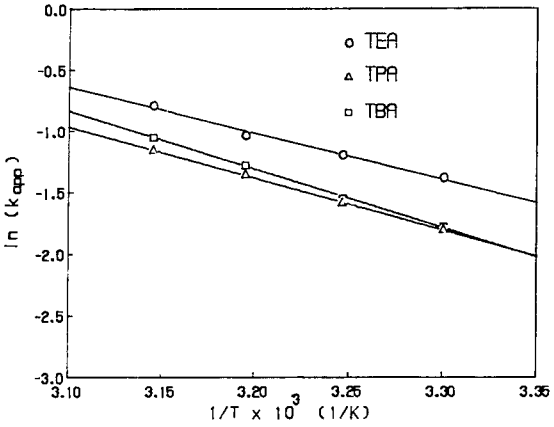


Fig. 6. Arrhenius plot of $\ln(k_{app})$ vs. 1/T for the reaction using varous tertiary amines; 0.4 g of $C_6H_4(NH_2)_2$, 4.003 g of CS_2, 1.68×10^{-3} mole of tertiary amine, 0.8188 g of KOH, 50 mL of chlorobenzene, 20 mL of H_2O, 1000 rpm

The activation energy for using those catalysts are 31.56 ~ 40.02 KJ/more for which TEA possesses the lowest value.

V. CONCLUSION

The kinetics of the reaction of o-phenylene diamine and carbon disulfide catalyzed by tertiary amine in the presence of KOH in a two-phase medium was investigated. A kinetic model, which includes the reaction of tertiary amine and carbon disulfide to produce an active intermediate, and the reaction of the active intermediate and o-phenylene diamine to produce MBI product, was built up. The pseudo-first-order kinetics, which were simplified

from the original kinetic model, can be used to satisfactorily describe the experimental data. The reaction was identitifed as taking place in the organic phase. The reaction was enhanced by adding tertiary amine and potassium hydroxide. The salt of the MBI product, which was obtained from the reaction of MBI and potassium hydroxide at the organic-aqueous interface, dissolved in the aqueous phase. The MBI product precipitates from the aqueous solution by adding an acidic substance and can be easily separated from the aqueous solution by filtration or centrifuge.

ACKNOWLEDGMENT

The authors would like to thank the National Science Council of the ROC for the financial support of this manuscript under contract no. NSC-83-0402-E-007-004.

REFERENCES

1. Blocher, S.H., A. Schultze and H.W. Leverkusen, "Process for the Production of 2-Mercaptobenzimidazole," U.S. Patent 3,235,559 (1966).
2. Broda, W. and E.V. Dehmlow, "Thioharnstoffe Durch Dreikom Ponentenreaktionen Von Amien," *Liebiys Annual Chemistry*, pp. 1837-1843 (1983).
3. Dalgaard, L., L. Jaensen and S.O. Lawesson, "Synthesis, Rearrangements, and Fragmentation of Ketene Mercaptals Derived From Ketones or β-Diketones and Carbonn Disulfide," Tetrahedron, 30, pp. 93-104 (1974).
4. Dehmlow, E.V. and S.S. Dehmlow, *Phase Transfer Catalysis*, VCH Publishers, Inc., New York (1993).
5. Freedman, H.H., "Industrial Application of Phase Transfer Catalysis (PTC): Past, Present and Future," *Pure Apply and Chemistry*, 58, pp. 857-868 (1986).
6. Goodman, A.L., "Therapeutical Composition Containing 2-Mercaptobenzimidazole," U.S. Patent 3,558,775 (1971).
7. Keller, W.E., *Phase-Transfer Reactions*, Fluka-Compendium, Vol. 1 (1986).
8. Keller, W.E., *Phase-Transfer Reactions*, Fluka-Compendium, Vol. 2 (1987).
9. Liu, B.L., "A Study of Catalyzed Reaction of Synthesizing of 2-Mercaptobenzimidazole and Its Derivatives," Ph.D. Thesis, Department of Chemical Engineering, National Tsing Hua University, Hsinchu, Taiwan (1995).
10. Moreira, J.C., C.P. Luiz and G. Yoshitaka, "Adsorption of Cu (II), Zn(II), Hg(II) and Pd(II) from Aqueous Solutions on a 2-Mercaptobenzimidazole Modified Silica Gel," *Mirkrochimic Acta*, II, pp. 107-115 (1990).
11. Saxena, D.B., R.K. Khajuria and O.P. Suri, "Synthesis and Spectral Studies of 2-Mercaptobenzimidazole Derivatives. I.," *Journal Heterocyclic Chemistry*, 19, pp. 681-683 (1982).
12. Scherhag, B., E. Koppleman and H. Wolz, "Production of 2-Mercaptobenizimidazole by Reacting o-Phenylene Diamine and Carbon Disulfide," U.S. Patent 3,842,098 (1974).
13. Starks, C.M., "Phase Transfer Catalysis. An Overview," *American Chemical Society*, Symposium Series, 326, pp. 1-7 (1985).
14. Starks, C.M. and C. Liotta, Phase Transfer Catalysis, Principles and Techniques, Academic Press, New York (1978).
15. Thomas, C., "Hydrocarbon Oil Containing 2-Mercaptobenzimidazole," U.S. Patent 2,642,396 (1953).
16. Van Allan, J.A. and B.D. Deacon, "2-Mercaptobenzimidazole," *Organic Synthesis*, pp. 569-571.
17. Weber, W.P. and G.W. Gokel, Phase Transfer Catalysis in Organic Synthesis, Springer-Verlag, New York (1977).
18. Xue, G., X.Y. Huang, J. Dong and T. Zhang, "The Formation of an Effective Anti-corrosion Film on Copper Surfaces from 2-Mercaptobenzimidazole Solution," *Journal Electroanalytical Chemistry*, 310, pp. 139-148 (1991).
19. Yoshinori, T., N. Hajime, K. Chizuko, M. Toshyuk and A. Akira, "A New Route of 1, 2-Dithiole-3-Thiones by the Reaction of Enaminones with Carbon Disulfide," *Heterocycles*, 31, pp. 1-4 (1990).

Discussions of this paper may appear in the discussion section of a future issue. All discussions should be submitted to the Editor-in-Chief.

Manuscript Received: June 27, 1997
Revision Received: Oct. 15, 1997
and Accepted: Dec. 23, 1997

在氫氧化鉀存在下以三級胺催化二硫化碳和鄰苯烯二胺之反應

劉炳郎　王茂齡

國立清華大學化工系

摘　要

本研究主要以三級胺爲觸媒，在氫氧化鉀存在下，進行二硫化碳和鄰苯烯二胺在水溶液和有機溶劑之二相反應。在氫氧化鉀存在下，添和少量之三級胺就能大大地促進化學反應之進行。而合成 2- 硫醇基苯駢咪唑 (2-Mercapto-benzimidazole)之反應發生於有機相反應。然而，2-硫醇基苯駢咪唑鹽卻產生於二氯化碳和水之界面，然後再溶於水溶液中。使用此種方法合成產物最大之優點在於藉著添加酸性化合物於水溶液中便可得到固體晶將之2-硫醇基苯駢咪唑產物。由實驗數據得知，我們得以提供此反應之機構。反應乃由二硫化碳和三級胺首先進行產生一個活性中間體(R_3N-CS_2)，活性中間體再與鄰苯烯二胺進行反應合成 2- 硫醇基苯駢咪唑。除此，由於添加之氫氧化鉀與反應副產物H_2S進行反應得以使反應加速進行，而二硫化碳與鄰苯烯二胺之二相反應可以由擬一階反應速率定律來描述。

關鍵詞：合成 2- 硫醇基苯駢咪唑 (MBI) 三級胺，氫氧化鉀。

A TABU-SEARCH BASED ALGORITHM FOR CONCAVE COST TRANSPORTATION NETWORK PROBLEMS

Shangyao Yan* and So-Cheng Luo
Department of Civil Engineering
National Central University
Chungli, Taiwan 320, R.O.C.

Key Words: Concave Cost, Transportation Problem, Tabu Search.

ABSTRACT

This research employs the tabu search method to develop an algorithm for efficiently solving concave cost transportation network problems which are characterized as NP-hard. An initial solution method and a linear approximation approach are also developed, to evaluate the algorithm. The preliminary results show that the algorithm is potentially useful.

I. INTRODUCTION

The general transportation problem is formulated as a bipartite network, $G=(N,A,M)$, where N is the set of all supply nodes, M is the set of all demand nodes, and A is the set of all arcs. Arc (i, j) denotes the arc from supply node i to demand node j. Let S_i be the supply for node i in N, let D_j be the demand for node j in M and let $f(x_{ij})$ be the cost function for transporting x_{ij} amount of goods along arc (i, j). Also let u_{ij} and l_{ij} be the flow's upper bound and lower bound respectively for arc (i, j). The transshipment problem can be formulated as follows:

$$\text{Minimize } f(x)=\sum_i\sum_j f(x_{ij}) \qquad \text{(1a)},$$

$$\text{subject to } \sum_j x_{ij}=S_i \quad \forall i\in N \qquad \text{(1b)} \qquad (1)$$

$$\sum_i x_{ij}=D_j \quad \forall j\in M \qquad \text{(1c)}$$

$$u_{ij}\geq x_{ij}\geq l_{ij} \quad (i,j)\in A \qquad \text{(1d)}.$$

The objective, (1a), minimizes the total transshipment cost. Constraints (1b) and (1c) are the flow conservation constraints at every supply node and at every demand node, respectively. Constraint (1d) ensures that every arc flow is within its upper and lower bound. There are $|N|\times|M|$ arcs in the network. As well, the sum of total supply is equal to that of total demand.

To facilitate problem solving, $f(x_{ij})$ is usually simplified as linear (for example, $f(x_{ij})=c_{ij}x_{ij}$), where there are numerous efficient algorithms for optimization, for example, the transportation network simplex method [1]. However, such linear cost functions may not reflect actual operations. In practice, the unit cost for transporting goods usually decreases as the amount of goods increases. For example, the fare structure for freight transportation or the cost function of garbage collection is generally concave [4].

Because concave cost transportation problems are characterized as NP-hard [13], and it is time-consuming to optimally solve large-scale problems. Many solution algorithms have been developed for handling such problems. These algorithms have employed traditional mathematical programming techniques such as linear approximation, Lagrangian relaxation, branch-and-bound, or dynamic programming to assist in their solutions [3, 4, 10, 18, 13]. Note that Zangwill [18] has pointed out that the major difficulty in solving concave cost network flow

*Correspondence addressee

problems results from the enormous local optima in the optimization. Thus, the traditional approaches may be inefficient for enumerating all local optima to find the global optimum. In our research we observed that the tabu search method (TS) [5, 6] may be useful for resolving concave cost transportation problems. For an example, see Problem (1).

In particular, any feasible spanning tree in Problem (1), corresponds to an extreme point of the constraint set [4]. Since every arc cost function is concave, the objective function (1a) is also, obviously, concave. Based on the concavity of the cost function, there is at least an optimal solution which is located at an extreme point. Furthermore, every extreme point is an integer solution because of the integrality property of Problem (1) , assuming that all the parameters in Problem (1) are integers. As a result, the optimal solution for Problem (1) is also an integer.

From this, we see that a traditional local search seems to be useful for finding the optimal solution for problem (1). That is, if we improve the solution from an extreme point to its adjacent extreme points, then, after a certain number of improvements, we might find a near-optimal solution. Unfortunately, since there could be enormous local optima in the solution set, such a local search could easily fall into a local optimum, whose objective is far from the optimal one. Recently there have been new combinatorial optimization approaches, such as TS, developed for efficiently "jumping" out of local optima. Therefore, TS could be applicable to the resolution of concave cost network flow problems.

TS considers all moves both "up" and "down" except for a certain prohibited or "tabu" set. The "tabu" moves are kept as a list of length L which effectively prevents the most recent L moves from being reversed. Each time a move is made, its "inverse" is added to the list, while the oldest move on the list is dropped. If the length, L, is too small, there is a chance that the method will simply cycle around the same sequences indefinitely, but this seems not to occur if the L is large enough [15]. This can be thought of as simulating a form of "short-term memory", so the procedure will recognize (and avoid) areas of the solution space that it has already encountered. Glover [5] also discussed ways of simulating "long-term memory" and procedures for overriding the basic algorithm by using "aspiration levels". The approach is well documented by Glover [5, 6].

TS has been successfully applied to many problems. For example, Sinclair [16] has used real world data to compare TS with several other modern heuristics for the problem of balancing hydraulics (a special case of the quadratic assignment problem). The results showed that TS provided the best solutions,

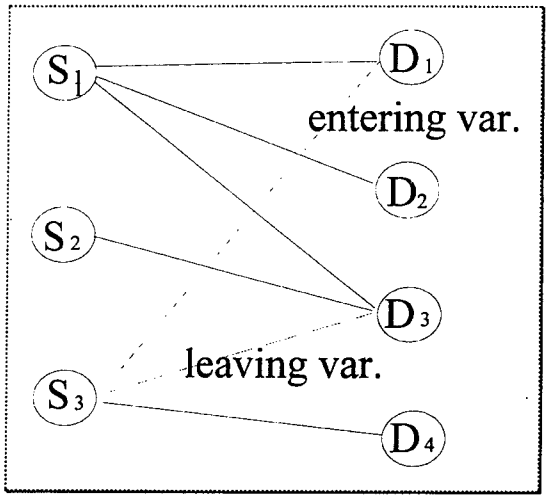


Fig. 1. A local move for a spanning tree

but at the cost of long solution times. Reeves [15] applied TS to the machine sequencing problem and found that TS is a more effective search paradigm than simulated annealing. Hu [9] applied TS to minimum weight design examples of a three-bar truss, coil springs, a Z-section and a channel section. The results showed that TS with random moves was a powerful approach to various problems of the global optimization of continuous variables. It outperformed the composite genetic algorithms and the random search for test problems with continuous variables.

This research aims to use TS to develop an algorithm that efficiently solves concave cost network flow problems. To reduce the complexity of this research, we focus on the concave cost transportation network problems, as shown in Fig. 1. Referring to [13, 14], the cost function for arc (i,j) used is assumed to be $f_{ij}(x_{ij})=c_{ij}\sqrt{x_{ij}}$, where x_{ij} denotes the flow of arc (i,j) and c_{ij} is an associate constant. Moreover, since constraint (1d) can be transferred into a nonnegativity constraint without upper bound constraints, then for simplification we use nonnegativity constraints to stand for constraint (1d). The remainder of the paper is organized as follows: we first develop solution algorithms, then perform a computation test, and finally conclude.

II. ALGORITHM DEVELOPMENT

In this section, we develop an initial solution method, a TS-based algorithm and a linear approximation approach.

1. Initial solution (INIT)

Since a local improvement method must start

from a feasible solution, before introducing the TS-based algorithm, we develop a heuristic to generate good initial solutions. Based on the characteristics of concave cost network flow problems [12], if arc flows are assigned for nodes with more supply or demand, the cost might be further reduced. Besides, an all-or-nothing assignment rule might help reduce the transshipment cost. The heuristic based on these is developed as follows.

Step 1. Sort all arcs;

(1.1) Determine the maximum amount of goods, x_{ij}, that can be transported from supply node i to demand node j; $x_{ij}=min(S_i, D_j)$.

(1.2) Calculate the unit cost for transporting x_{ij} which is equal to $\frac{c_{ij}}{\sqrt{x_{ij}}}$;

(1.3) Sort all arcs increasing order, according to $\frac{c_{ij}}{\sqrt{x_{ij}}}$;

Step 2. Assign arc flows;

(2.1) Assign arc flows sequentially using the remaining node supply and demand;

(2.2) Calculate the remaining supply or demand for all nodes;

Step 3. If all node supply and demand have been assigned, then go to **Step 4**; otherwise go to **Step 1**;

Step 4. If the number of arcs with positive flows is less than ($|M|+|N|-1$), then add arcs with zero flow to form a spanning tree; Stop the algorithm;

Note that in each iteration, the supply or demand for at least half of the nodes with remaining supply or demand will become zero. Let n be $|M|+|N|$ and m be $|M|\times|N|$. Thus, the INIT is finished in at most $\log_2$ n iterations. Since the complexity of each iteration is O(m*log m) when using the heap sort approach in Step 1, the complexity of the INIT is O(m×log m×log n).

2. The TS-based algorithm (TSA)

The development of the tabu-search based algorithm is addressed in the following three parts.

(1) Neighborhood searching

We refer to the pivoting rules of the network simplex method [1], to search for adjacent extreme points. In particular, given an initial spanning tree (an extreme point), an "entering arc" incorporating the spanning tree forms a unique circuit. A "leaving arc" can then be determined and deleted to form another spanning tree, which is an adjacent extreme point to the previous one [1]. Thus, a pair arcs of entering and leaving can be indicated as the direction of a move. As shown in Fig. 1, arc (3,1) is the entering arc and arc (3,3) is the leaving arc. Note that in the network simplex method the entering arc is chosen based on a negative reduced cost [1].

Given a spanning tree, there are many arcs that can be chosen as entering arcs (a total of m-n+1 arcs). Two neighborhood searches have always been used. One is the steepest descent search and the other is the random search. To perform the former we have to calculate the flow difference for every adjacent point in order to choose the best one. Thus, we have to identify all associated circuits and changes in the associated arc flows. This, as a result, will be time-consuming, especially when the calculations involve a complex concave cost function. Although the latter may provide a faster search, the solution quality of the moves may be ineffective. To tradeoff efficiency and effectiveness, we suggest examining a limited number (denoted as EPOCH) of arcs in every move. For simplification, EPOCH is suggested to be a multiple of the number of total nodes, where the multiple is subject to testing. We randomly choose an EPOCH of arcs from all feasible arcs and then determine the entering one. The following pseudo code addresses the searching process.

Procedure SEARCH
Begin
k=1;
Δ=9999999999;
Do while (k<EPOCH)
  Randomly choose a new arc which has not been chosen;
  Identify the associated circuit and find the associated leaving arc from the circuit;
  Modify the arc flows and calculate the associated difference of the objective value, Δ_k;
  If $\Delta_k<\Delta$, then $\Delta=\Delta_k$, save the new solution as an incumbent;
  $k=k+1$;
End do
End_SEARCH.

(2) The tabu lists and the aspiration levels

Given an initial spanning tree, we use tabu lists to avoid inverse or inefficient moves. Two kinds of tabu lists are proposed here. One is called an "inverse tabu list" (ITL) and the other is called a "degenerate tabu list" (DTL). The former prevents a list of recent moves from being reversed, while the latter prevents a list of degenerate moves from being performed. To suitably define the ITL, we propose a new approach for effectively recording and distinguishing recent moves. In particular, we use random

numbers at the beginning to generate several random parameters for every arc. Then we define several random objective functions, as shown in Eq. (2).

Original objective function

$$z(x)=\sum_i\sum_j f_{ij}(x_{ij})=\sum_i\sum_j c_{ij}\sqrt{x_{ij}} \qquad (2a)$$

Random objective function 1

$$z_1(x)=\sum_i\sum_j f^1_{ij}(x_{ij})=\sum_i\sum_j P^1_{ij}x_{ij} \qquad (2b) \qquad (2)$$

Random objective function 2

$$z_2(x)=\sum_i\sum_j f^2_{ij}(x_{ij})=\sum_i\sum_j P^2_{ij}x_{ij} \qquad (2c)$$

Random objective function s

$$z_s(x)=\sum_i\sum_j f^s_{ij}(x_{ij})=\sum_i\sum_j P^s_{ij}x_{ij} \qquad (2d)$$

If every objective function value is the same for two solutions, then these two solutions are likely to be the same. In other words, if all the objective function values for a new solution are the same as that of any solution in the ITL, we reject this move unless it satisfies the aspiration levels, which will be mentioned later. Otherwise, we accept this move and also add its objective function values to the ITL. Obviously, the accuracy for identifying a recent move will depend on the number of random objective functions that we have defined. Generally, five objective functions are enough for our problems. Note that the same ITL length is maintained for every move. The performance of the length, as mentioned in [5, 6], is subject to testing.

For a spanning tree, there are always arcs with zero flow (called degenerate arcs) due to degeneracy. It is believed that network flow problems are usually degenerate [17]. Thus, when selecting an entering arc for determining a move, if the associated leaving arc is a degenerate arc, then this move will "stand still", that is, nothing will be changed in the new solution. If the problem is highly degenerate, then although many moves seem to have been performed, the new solution has in fact stood still throughout. Such a degeneracy problem could cause severe deterioration in the solution efficiency. At worst, it could result in an infinite algorithm. To resolve this, we use the DTL to record recent degenerate arcs. When we search for a new entering arc, we check to see if it is in the DTL. If it is, we reject this arc; otherwise we accept it as a candidate for an entering arc. If the arc is examined and found to be degenerate later, then we add it to the DTL. Note that a degenerate arc in the DTL is deleted from the DTL after a certain number of moves, which are subject to testing.

To provide flexibility for choosing good moves, referring to [5, 6] we set an aspiration level (AL). The AL allows us to override a tabu move if the objective value of the new solution is better than that of the previous one. Note that to fasten the searches, we apply the AL only to the ITL rather than to the DTL, because the objective values do not need checking in advance if an arc is located in the DTL. To efficiently incorporate the DTL into a neighborhood search, we modify the SEARCH as the MSEARCH.

Procedure MSEARCH
Begin
k=1; Δ=9999999999;
Do while (k<EPOCH)
  Randomly choose a new arc which has not been chosen and is not in the DTL;
  Identify the associated circuit and find the associated leaving arc from the circuit;
  Modify the arc flows and calculate the resulting difference of the objective value, Δ_k;
  If the arc flows are not changed, then add the entering arc into the DTL; if $\Delta_k<\Delta$, then $\Delta=\Delta_k$ and save the new solution as an incumbent;
  $k=k+1$;
End do
End_MSEARCH.

The complexity for identifying a circuit and changing its flow, given a candidate entering arc, is $O(m)$. Let ld be the maximum length of the DTL. The complexity of selecting a candidate entering arc is O(ld). As a result, the complexity of MSEARCH is $O(EPOCH*(m+ld))$

(3) The TSA procedure

Let S be the current solution, S' be the new solution, S_{inc} be the incumbent, and MAX be the maximum number of moves for which an incumbent has not been improved. Referring to [5, 6] and the aforementioned issues, we develop a TS-based algorithm as follows.

Step 0. Use the INIT to find an initial solution, S, and calculate its objective value, $z(S)$, and all random objective function values, $z_1(x)$, ..., $z_s(x)$; $S_{inc}=S$; count=0;

Step 1. Apply the MSEARCH to find a good adjacent point, S'; Calculate its objective function value, $z(S')$ and its random objective function values, $z_1(S')$,..., $z_s(S')$;

Step 2. If S' is not in the ITL or if $z(S')<z(S)$, then $S=S'$, update ITL and DTL; otherwise count=count+1 and go to **Step 4**;

Step 3. If $z(S')<z(S_{inc})$, then $S_{inc}=S'$ and count=0;

otherwise count=count+1;

Step 4. If count>MAX, then stop; otherwise go to **Step 1**;

We note that the TSA is finite. First the INIT is finite. Then, every incumbent is obviously an extreme point. An incumbent is changed (is improved) in at most MAX iterations. Since the number of extreme points in Problem (1) is finite which is bounded by C^m_{n-1}, the number of incumbent solutions is finite. As a result, the TSA finishes in a finite number of iterations. Moreover, the length of ITL is typically less than $O(EPOCH\times(m+ld))$. Omitting INIT whose complexity is relatively small, the complexity of TSA is then equal to $O(MAX\times C^m_{n-1}\times EPOCH\times (m+ld))$. Although this complexity is exponential, the real performance in practice is typically better than the complexity shows [5, 6].

3. Linear approximation heuristic (LAH)

To preliminarily evaluate the TSA, we develop a linear-cost approximation heuristic [10]. Note that the heuristic was shown to be a good method by Jordan [10] and is particularly easy to code for solving our problems. The heuristic starts from an initial solution which can be obtained using SEARCH. In each iteration the heuristic uses the network simplex method to optimally solve for a linear cost transportation problem. The cost function of the transportation problem is estimated using the arc flows of the previous solution. For example, if the flow from arc (i, j) in the previous solution is x_{ij}', then the estimated cost function for this arc is

$$f(x_{ij})=[f(x_{ij}')/x_{ij}']\times x_{ij}=\frac{c_{ij}x_{ij}}{\sqrt{x_{ij}'}} \quad (3)$$

where x_{ij} is the arc flow in this iteration. The process is repeated until an incumbent has not been improved for 1000 iterations. The steps for the LAH are listed below. The reader can similarly prove that the LAH is finite as was found in the TSA.

Step 0. Use the INIT to find an initial solution, S, and its objective value, $z(S)$; $S_{inc}=S$; count=0;

Step 1. Approximate a linear cost function using S;

Step 2. Solve the modified linear cost transportation problem and obtain a new solution S';

Step 3. If $z(S')<z(S_{inc})$, then $S_{inc}=S'$ and count=0; else count=count+1;

Step 4. If count>1000, then stop; otherwise $S=S'$ and go to **Step 1**;

Note that the complexity of solving the modified linear cost transportation problem is $O(C^m_{n-1})$, although it usually took a few pivots to find the optimal solution, given the previous optimal basis. Let max be the maximum number of iterations allowed for holding an incumbent. Similar to TSA, omitting INIT, the complexity of LAH is $O(max\times C^m_{n-1}\times C^m_{n-1})$. Although this complexity is higher than that of TSA, the real performance of LAH is not as poor as the complexity shows [10]. This will also be shown in next section.

III. COMPUTATION EXPERIMENTATION

In this section, we will discuss how we designed a network generator, determined parameters for the TS-based algorithm, and evaluated the preliminary results.

1. Network generator

To test the algorithms we designed a network generator to generate concave cost transportation network problems. Since the performance of the TSA may be influenced by problem scales and parameters, we used random numbers to generate randomized networks with various network scales and parameters [11]. We first randomly set the number of supply nodes and demand nodes, then randomly set the supply and demand for all nodes. To ensure that the flows were balanced at all nodes, the sum of all node supply was set equal to that of all node demand. Thereafter, we built all arcs between every supply node and every demand node. The arc cost parameters were randomly set to be positive. We note that there was at least one feasible solution for any randomized network. In addition, since the arc cost functions were positive, the objective function for each network was bounded from zero. As a result, there was an optimal solution for every network.

2. TSA parameters

Since the TSA is a metaheuristic, its performance is related to its parameters [5, 6]. Before evaluating the TSA and other algorithms, we use the tested networks in the next section **(3. Preliminary results)** as input to search for good parameters for the TSA. After numerous tests, we found that, generally, the larger the MAX, the better the objective value; but the longer the computation time. However, when MAX increases to more than 1500, the solution quality is not significantly improved.

Having performed a sensitivity analysis, we found that the best ITL length is 7, which is not significantly different from 6 or 8. We also found that when the ITL length was more than 8 (it was more time-consuming for a move), the total computation time was longer, while the final objective was not

Table 1. Computational results of small-scale networks

	INIT			LAH			TSA			optimal
Network	obj	%obj	time	obj	%obj	time	obj	%obj	time	obj
4×4	1416.12	56.67	0.02	1155.30	27.81	0.02	903.89	0.00	0.85	903.89
4×5	1369.62	16.54	0.04	1236.15	5.18	0.04	1175.26	0.00	2.08	1175.26
5×6	2140.01	37.44	0.00	1987.56	27.65	0.04	1557.04	0.00	2.03	1557.04
6×5	1702.50	25.36	0.02	1654.31	21.81	0.16	1358.09	0.00	3.92	1358.09
7×4	1471.91	1.58	0.00	1471.91	1.58	0.00	1448.97	0.00	5.58	1448.97
average	1620.03	27.52	0.02	1501.05	16.81	0.05	1288.65	0.00	2.89	1288.65

Note: %obj=100×(obj−optimal obj)/optimal obj

significantly better. On the other hand, when the ITL length was less than 6, although each move took less time, the total computation time was still longer, probably due to many ineffective moves. Besides, the final objective was not superior.

We found that the DTL is useful for improving solution efficiency. In particular, it can avoid ineffective selections of entering arcs by choosing a suitable number of moves (denoted as NOM) for which a degenerate arc is retained in the DTL. From our results, a number equal to log (n) is the best for setting the NOM. Having performed a sensitivity analysis, we found that if the NOM was too small (less than log (n)), then many degenerate moves typically occurred, while, if the NOM was too large (larger than log (n)), then the flexibility for choosing entering arcs was significantly reduced, resulting in an ineffective searching of neighbors. As a result, when NOM was more than or less than log (n), the computation time was longer, while the final objective was not superior.

Besides, in most cases if two solutions were found to have the same objective function value, then they were degenerate solutions. Very few of them were different solutions. From this, the number of random objective functions designed for the ITL may be suitably reduced to avoid superfluous calculations. After all, the TSA parameters are suggested to be as follows: the length of the ITL=7, the length of the DTL is not limited (at most m), NOM=log(n), EPOCH=n, MAX=1500.

Note that searching of the above parameters in this research was typically by a trial-and-error process. For example for the setting of MAX, we first fixed other parameters, then set different values of MAX from 1000 to 2500 with an increment of 100 to decide the best value. Similarly, we changed another parameter and tried to set the best MAX as above. The process was repeated until we found the best combination of all parameters. Obviously, this way is very time-consuming and inefficient especially for testing large-size problems. There could exist a certain relationship between a given problem instance and its associated parameters. Thus, how to explore such a relationship in order to find an efficient and effective way to determine the parameters for the application of TS in practice could be a direction of future research.

3. Preliminary results

Based on the aforementioned parameters, we evaluated INIT, TSA and LAH on an HP735 workstation. We first generated five small-scale networks. These small-scale problems can be manually optimized. As shown in Table 1, the average error for the INIT was 27.52%. The TSA performed well and optimized the initial solutions to all problems. Although the LAH improved the initial solution, the average error (16.81%) was still significant, meaning that the LAH was not as effective as the TSA for solving the problem instances. As for computation times, both INIT and LAH were efficient. Both were finished in an average of 0.05 seconds. The TSA was relatively slow, with an average of 2.89 seconds.

To better understand the performance of these algorithms, we randomly generated fourteen other networks with larger-scale sizes. The results are summarized in Table 2, where the error percentage of the objective values is calculated based on the best value obtained from among all the algorithms. The average error of objective values for INIT was 83.56%. In the fourteen networks, the LAH was worse than the TSA, with an average of 71.28% error relative to the TSA.

As for computation times, they are positively correlated with problem sizes for all algorithms. That is, the computation time increases with problem size for every algorithm. The INIT was finished in an average of 42.58 seconds. Although the complexity of TSA is lower than that of LAH, the TSA was l onger than the LAH, the former being finished in an average of 198.61 seconds, while the latter was finished in an average of 44.61 seconds. Based on the above results, if we do not count the computation

Table 2. Computational results of small-scale networks

	INIT			LAH			TSA			best	
Network	obj	%obj	time	obj	%obj	time	obj	%obj	time	obj	method
10×10	1844	55.22	0.04	1742	46.63	0.54	1188	0	5.74	1188	TSA
12×12	2331	22.30	0.04	2222	16.58	0.50	1906	0	4.80	1906	TSA
20×20	2253	49.80	0.12	2018	34.18	1.34	1504	0	10.18	1504	TSA
25×25	64797	54.44	0.22	59862	42.68	2.20	41955	0	40.22	41955	TSA
30×30	3498	103.25	0.40	3029	76.00	4.38	1721	0	23.40	1721	TSA
40×40	4260	86.35	0.94	4015	75.63	3.84	2286	0	52.70	2286	TSA
50×50	3466	67.93	1.86	3279	58.87	5.62	2064	0	72.96	2064	TSA
80×80	5069	73.12	8.80	4987	70.32	10.72	2928	0	62.70	2928	TSA
80×120	5713	77.53	20.62	5533	71.94	30.88	3218	0	75.60	3218	TSA
100×100	133366	119.83	20.04	126067	107.80	25.02	60667	0	183.26	60667	TSA
150×150	165802	134.65	38.50	158769	124.70	48.92	70659	0	335.78	70659	TSA
110×220	183335	107.95	129.48	176221	98.74	140.86	88165	0	624.82	88165	TSA
180×160	175061	97.10	130.56	160113	80.27	151.44	88817	0	312.30	88817	TSA
200×200	173453	120.39	244.46	152396	93.63	198.28	78704	0	976.08	78704	TSA
average	66018	83.56	42.58	61447	71.28	44.61	31842	0	198.61	31842	

Note: %obj=100×(obj-best obj)/best obj

times, which differ less than 160 seconds for the demonstrated computation capability, then the TSA is apparently better than the LAH.

Note that the greatest difference in computation time among all instances, whose scales are close to practical ones (for example, 200×200), is less than 13 minutes which is not crucial in practice. It should be mentioned that although TSA is less efficient than LAH, TSA should be more applicable than LAH in practice. The reason is that the time for solving transportation problems in the planning stage is generally not constrained in practice. Carriers usually have plenty of time to solve problems in advance. Even if the time is constrained in special events, then more powerful computers could be used to increase the speed of problem solving. In the future, various computers can be used to test larger size problems than the ones tested in order to understand the range of problem sizes that can be practically solved.

IV. CONCLUSIONS

This research employed the tabu search method to develop a TS-based algorithm to efficiently solve concave cost transportation network problems. To preliminarily evaluate the algorithm, a linear approximation heuristic (LAH) was developed. A heuristic (INIT) for generating initial solutions was developed based on the problem characteristics. A network generator was also designed to generate many instances on an HP735 workstation to test the heuristics. The results show that although the TSA is computationally longer than the LAH, the TSA apparently outperforms the LAH in terms of objective values. In summary, the preliminary results show that the TSA is potentially useful for solving concave cost transportation network problems.

It should be mentioned that the above evaluation of the TSA was based on the parameters proposed in the section on Computation Experimentation. Since the performance of the TSA may be influenced by problem sizes and other parameters, more tests on larger-size problems and algorithm parameters should be performed in the future to further evaluate the algorithm, so that it can be effectively applied in practice. In addition, how to find an efficient and effective way to determine the parameters for the application of TS in practice could be a direction of future research. The evaluation of other concave cost functions for transportation network problems or other network problems could also be directions of future research. Finally, we note that the simulated annealing method [8], the threshold accepting method [2] and the generic algorithm [7] have all been shown to be good for solving certain combinatorial optimization problems and could be useful for resolving concave cost network flow problems as well. They are also suggested as future areas of research.

ACKNOWLEDGMENTS

This research was supported in part by a grant (NSC-87-2211-E-008-013) from the National Science Council of Taiwan. We thank the three anonymous referees for their valuable comments and suggestions.

NOMENCLATURE

A	set of all arcs
AL	aspiration level
c_{ij}	unit cost for transporting goods along arc (i, j)
D_j	demand for node j in M
DTL, ITL	degenerate and inverse tabu list respectively
$EPOCH$	number of neighborhood search
$f(x_{ij})$	cost function for transporting x_{ij} amount of goods along arc (i, j)
$INIT$	Initial solution
k, Δ, Δ_k, p^1_{ij}, p^2_{ij}, p^3_{ij}	count parameters
L	length of tabu list
LAH	Linear approximation heuristic
ld	maximum length of DTL
l_{ij}, u_{ij}	flow's lower and upper bounds respectively for arc (i,j)
M, N	set of all demand and supply nodes
m, n	equals $\|M\|\times\|N\|$ and $\|M\|+\|N\|$
MAX, max	maximum moves in TSA and LAH for which an incumbent has not been improved
NOM	number of moves for which a degenerate arc is retained in the DTL
obj	objective function value
S, S^*, S_{inc}	current, new and incumbent solutions
S_i	supply for node i in N
x_{ij}, x'_{ij}	flow of arc (i,j) now and in the previous iteration respectively
TS	tabu search
TSA	TS-based algorithm
$z(x)$, $z_1(x)$, $z_2(x)$, $z_s(x)$	objective functions
(i, j)	arc from supply node i to demand node j

REFERENCES

1. Ahuja, R.K., T.L. Magnanti and J.B. Orlin, *Network Flows, Theory, Algorithms, and Applications*, Prentice-Hall (1993).
2. Dueck, G. and T. Scheuer, "Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing," *Journal of Computational Physics*, Vol. 90, pp. 161-175 (1990).
3. Gallo, G., C. Sandi and C. Sodini, "An Algorithm for the Min Concave Cost Flow Problem," *European Journal of Operational Research*, Vol. 4, pp. 248-255 (1980).
4. Gallo, G. and C. Sodini, "Adjacent Extreme Flows and Application to Min Concave Cost Flow Problems," *Networks*, Vol. 9, pp. 95-121 (1979).
5. Glover, F., "Tabu Search, Part I," *ORSA Journal on Computing*, Vol. 1, pp. 190-206 (1989).
6. Glover, F., "Tabu Search, Part II," *ORSA Journal on Computing*, Vol. 2, pp. 4-32 (1990).
7. Goldberg, D.E., *Genetic Algorithm in Search, Optimization and Machine Learning*, Addison-Wesley, Massachusetts (1989).
8. Golden, B.L. and C.C. Skiscim, "Using Stimulated Annealing to Solve Routing and Location Problems," *Naval Research Logistics Quarterly*, Vol. 33, pp. 261-279 (1986).
9. Hu, N., "Tabu Search Method with Random Moves for Globally Optimal Design," *International Journal for Numerical Methods in Engineering*, Vol. 35, pp. 1055-1070 (1992).
10. Jordan, W.C., "Scale Economies on Multi-Commodity Distribution Networks," GMR-5579, Operating Systems Research Dept., GM Research Laboratories (1986).
11. Klingman, D., A. Napier and J. Stutze, "NETGEN: A Program for Generating Large Scale Capacitated Assignment, Transportation, and Minimum Cost Flow Network Problems," *Management Science*, Vol. 20, pp. 814-821 (1974).
12. Kuhn, H.W. and W.J. Baumol, "An Approximate Algorithm for the Fixed-Charge Transportation Problem," *Naval Research Logistics Quarterly*, Vol. 9, pp. 1-16 (1962).
13. Larsson, T., A. Migdalas and M. Ronnqvist, "A Lagrangian Heuristic for the Capacitated Concave Minimum Cost Network Flow Problem," *European Journal of Operational Research*, Vol. 78, pp. 116-129 (1994).
14. LeBlanc, L.J., "Global Solutions for a Nonconvex, Nonconcave Rail Network Model," *Management Science*, Vol. 23, pp. 131-139 (1976).
15. Reeves, C.R., "Improving the Efficiency of Tabu Search for Machine Sequencing Problems," *Journal of the Operational Research Society*, Vol. 44, pp. 375-382 (1993).
16. Sinclair, M., "Comparison of the Performance of Modern Heuristic for Combinatorial Optimization on Real Data," *Computers and Operations Research*, Vol. 20, pp. 687-695 (1993).
17. Yan, S., "Approximating Reduced Costs under Degeneracy in a Network Flow Problem with Side Constraints," *Networks*, Vol. 27, pp. 267-278 (1996).

18. Zangwill, W.I., "Minimum Concave Cost Flows in Certain Networks," *Management Science*, Vol. 14, pp. 429-450 (1968).

Discussions of this paper may appear in the discussion section of a future issue. All discussions should be submitted to the Editor-in-Chief.

Manuscript Received: Sep. 17, 1997
Revision Received: Feb. 19, 1998
and Accepted: Mar. 03, 1998

禁制搜尋法於求解凹形成本運輸網路問題之研究

顏上堯　羅守正

國立中央大學土木工程學系

摘　要

本研究運用禁制搜尋法技巧，發展一演算法，以有效的求解凹形成本運輸網路問題。此網路問題在運算上可歸類爲NP-hard問題。在此研究中，我們亦發展一起始解法及一線性估計法，以評估求解的績效。目前結果顯示，本研究發展之禁制搜尋法的效果頗佳。

關鍵字：凹形成本、運輸問題、禁制搜尋法。

A DATABASE APPLICATION GENERATOR FOR THE WWW

Wei-Jyh Lin
Computer Science Department
National Chengchi University
Taipei, Taiwan 116, R.O.C.

Kung Chen*
Department of Information Management
National Taiwan University of Science and Technology
Taipei, Taiwan 106, R.O.C.

Key Words: WWW, CGI, RDBMS, ODBC, internet, intranet.

ABSTRACT

This paper describes a database application generator for the WWW called GWB. GWB contains a compact language that adds control structures and database access constructs to HTML, a compiler that translates HTML-like source templates into ODBC code and utilities for authentication and session management. It is designed to ease the expertise requirement needed for developing Web-based intranet and internet database applications. This paper surveys the current approaches; describes the language and its support for authentication and session management; and gives an internet application using GWB. This paper also discusses future enhancement in terms of persistent database connections and server-side client state persistency.

I. INTRODUCTION

The popularity of the World Wide Web (WWW) [2] has resulted in a trend to open up organization information, which has been accessible only internally, to the public. Increasingly more corporations are utilizing the WWW to distribute corporate information to external customers and to develop internal IS applications. Many are even providing electronic stores where product information can be requested on demand and items can be electronically purchased.

Most of this information is stored in a Relational Database Management System (RDBMS). Using the Web to develop database applications involves writing HTML [23] as the front-end user interface and Common Gateway Interface (CGI) [9] programs for back-end database access.

The mechanism that a Web client accesses a database is depicted in Fig. 1. Users click on an anchor in an HTML page which represents a user interface to databases. This action triggers the client to send to web servers a GET or POST HTTP [24] message specifying a CGI program to run with arguments from user input. The web server executes the gateway program which services SQL requests embedded in input arguments by connecting to the database server and sending the request for execution. The results are then passed back to clients along the same route, from database servers, to gateways, to web servers, and finally to clients. Other mechanisms of accessing databases from the Web exist. Please see section 2.2 below.

In the above CGI mechanism the database gateway program has to do several tasks: (1) it has to decode the parameters passed from a web server and validate them; (2) it has to compose SQL statements according to input values, including necessary binding of SQL parameters to host variables; (3) it has to

*Correspondence addressee

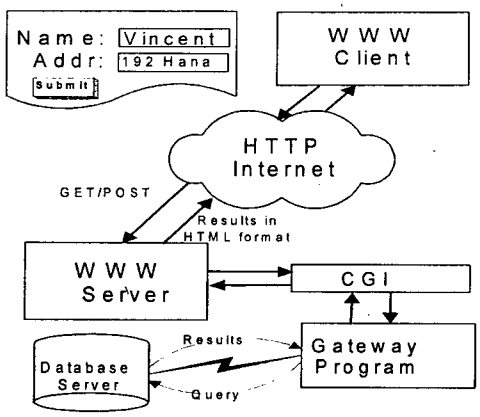


Fig. 1. The architecture of the CGI.

perform database connection, sending requests, examining execution status, and retrieving result sets if any; and finally (4) it has to formulate results back into HTML, which includes HTML-escaping database texts, numeric values, and blob data, before sending them back to clients.

All of the above tasks are tedious and repetitive when developing database programs the WWW. They are low level and should be automated. There is a clear need for database application generators to automate these tasks.

GWB is designed to automate these tasks. Fig. 2 depicts GWB architecture. GWB users write HTML-like source programs. GWB first compiles the source programs into ANSI C code. It then invokes a C compiler to compile the generated code and link with GWB and ODBC libraries to produce a database access gateway program. A separate HTML form can be authored using any WYSIWYG HTML editor to invoke the gateway program.

II. CURRENT APPROACH SURVEY

Since Arthur Secret's work on WWW access to relational databases at CERN in 1992, there has been a growing interest in this area. Individual researchers, database middleware vendors, and database vendors all have their own tools to address this issue. Richmond [16] and Rowe [18] maintain comprehensive lists of Web/DBMS tools and products. For the purpose of designing a database application generator we surveyed existing tools from two different perspectives: *programming model and architecture.*

1. Programming model

By *programming model* we mean the abstract model of a language in which applications are programmed. A language's programming model determines how naturally the intended Web database processing and presentation can be expressed. The language constructs and database access primitives determine how general applications can be written. We classified programming models of existing tools as *HTML-based, Perl-based, Script-based, and other-programming-language-based.*

HTML-based tools generally take HTML as a base language and add a few extended tags for application processing logic. A source program, called a *template*, is simply the intended HTML output with control constructs and SQL statements embedded. Special syntax is provided for variables that act as place holders for parameters passed from Web servers. Variables can be printed as HTML texts, incorporated in SQL statements, or used in branching or looping constructs. These tools also provide mechanism to iterate through and HTML-escape retrieved data before presenting them to clients.

This approach has the advantage of simplicity. A WYSIWYG HTML editor can be used to write the base HTML part of a template. Control structures can then be added. The output presentation of an application is thus clearly separated from its processing. Typical tools in this category include Cold Fusion [1], SWOOP [5], WebDBC [19], and DB2 WWW[13]. GWB also belongs to this category.

Tools in this category vary in how new tags and variables are introduced syntactically. They also differ in whether general operators and expressions are provided. Limited expressions constrain the scope of applications that can be written. Some tools do not allow multiple SQL statements in one template. Virtually none of these existing tools offer complete data types and database access primitives such as those provided by ODBC API.

The second category includes tools that use Perl as their programming language. Perl-based tools are among the first Web/RDBMS tools. Michael Peppler in Switzerland and Kevin Stock in the UK have developed sybperl [20] and oraperl [15], which are perl implementations of the C library routines for Sybase and Oracle databases. Since Perl has complete control constructs, arbitrary complex Web applications can be written using these libraries. As a result this approach has been very popular in publishing databases on the internet.

Although these tools provide all necessary primitives for writing database applications they are not specifically designed for Web/Database access. The presentation of application output typically disperses in Perl statements. Other tools, such as Sybase's web.sql [17] and WDB [10], removes this drawback by providing another layer's programming paradigm on top of these Perl libraries.

The third category contains script-based tools

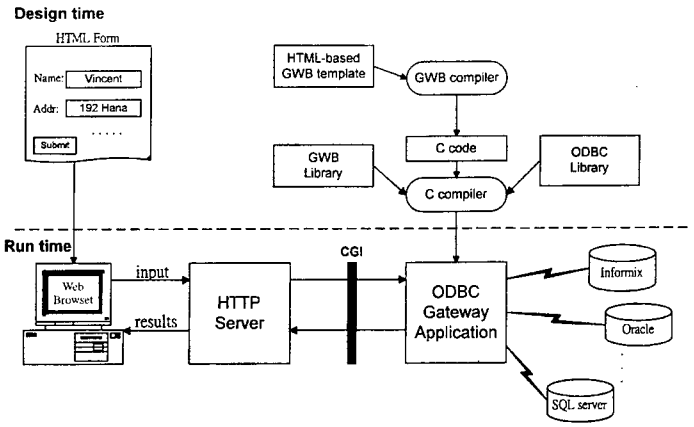


Fig. 2. GWB architecture.

that define their own specific script languages. These tools generally assume particular access patterns and presentation styles and optimize the script languages accordingly. For example, Web/Genera [7] assumes that most database access from the Web is to query information from a complex database. It thus provides a schema description language that states the information to be displayed, including data types, column and table name, etc. WDB is another example of a script-based Web/Database tool that uses *form definition files* to describe which tables and fields should be accessible through each query form. Writing applications using these tools involves writing script statements to access databases and format results.

Since these tools assume access patterns and presentation styles, applications that do not follow these patterns or styles would be awkward to implement. These script languages generally do not have primitive control constructs. This further limits the kind of applications that can be programmed. None of these tools provide database access primitives.

Other-programming-language-based tools use Java, C++, etc. as their programming languages. These tools share the same characteristics of Perl-based tools: complete language constructs and database access primitives. These tools, however, differ from Perl-based tools in that they generally have a visual development environment. They are mostly designed to support large scale internet applications and thus all provide mechanisms to maintain state or perform load-balancing. See the following section for tools in this category such as [4], [11], [22].

2. Architecture

By *architecture* we mean the execution mechanism and environment of generated applications. The most popular architectures are *CGI*, *server API*, *special web server*, and *multi-tier architecture*.

CGI is the simplest form of Web/Database applications. A CGI application is simply a program that is forked and executed from a web server. It can be written in any language and virtually every web server supports CGI. There are two approaches to using CGI architecture. In the first approach the Web/Database tool is an interpreter which is run by web servers as a CGI program. Applications, whose paths are passed in URL query strings, are executed by the interpreter. Template-based and Perl-based tools, such as DB2 WWW and WebDBC, fall into this category. In the second approach the Web/Database tool compiles an application from a source format to a CGI executable. GWB and Script-based tools fall within this category.

The performance for CGI-based architecture is generally not optimal because each time a CGI is requested it has to be forked and executed by web servers. When there are many requests arriving at the same time web servers will be overloaded with CGI programs.

FastCGI from Open Market, Inc. [14] addresses this problem by providing a network protocol library which implements the CGI specification. Instead of running an external process, web servers compiled with this library communicate with a FastCGI program using this network protocol for each client request. This solves the performance issue. The scalability issue is also solved because FastCGI applications can run on multiple hosts which could be different from the one running web servers.

Netscape [12] was the first to pioneer server *API architecture*. The goal is the same as FastCGI: to avoid running a separate process for each client request. However, server API architecture does not seek to preserve the CGI mechanism between web servers and application programs. Instead, application

programs are compiled as loadable modules, such as shared libraries and DLL's, that are loaded by, and linked with, web servers at run-time. The performance is improved because each client request becomes an internal function call within web servers. Web.sql falls within this category.

Although server API architecture improves application performance most APIs are complex and difficult to use. Since application code is linked with web servers, immature applications can corrupt servers. In addition, applications do not scale up well since they must run with web servers on the same host.

Using a special web server is another way to increase performance. Special web servers that can complete application requests, in addition to servicing HTML pages, are used in place of a regular web server. When a request arrives, special web servers determine if it is a request for an HTML page or for an application. In case of the latter they would execute the application as part of server functions without running a separate process.

Special web server architecture shares the same problems as server API based architecture: applications and web servers can interact in unpredictable ways, and the applications do not scale up well. A special web server approach is in general easier to use than the complex server API's.

Multi-tier architecture has become an increasingly important architecture recently. Multi-tier architecture involves web servers and a number of application servers and database servers working together in servicing requests. The application servers and database servers run as daemon processes and wait for requests from web servers. When a request arrives web servers pass the request to an available application server through a small CGI or server API. The application server would run a Java class or interpret a template or a script as specified by the request. For each database access embedded in the request the application server asks an available database server to access the desired data. The results are then processed by application servers before sending the data back to the client. NetDynamics [11] and dbKona/T3 [22] are representative of multi-tier architecture.

Multi-tier architecture solves performance and scalability problems simultaneously by employing multiple application and database servers. In addition, transaction, session, and state management can all be addressed in this architecture. Section five discusses future work of GWB towards a multi-tier architecture.

III. GWB DESIGN GOALS

An ideal Web/RDBMS tool should be easy to use without requiring specific programming expertise. It should be designed for both internet and intranet applications. Ideally, it should also scale up without sacrificing performance and ease-of-use. Specifically the design goals of GWB are to fulfill the following requirements.

1. Simplicity: Much of the tedious and low-level work should be simplified. For example, a simple mechanism should be provided to access input arguments without having to worry about decoding. Presenting results should be straightforward without having to worry about HTML-escaping of data and whether the piece-meal composition of HTML texts would produce the desired layout.
2. Power: Not only should simple client/server applications be easy to develop but writing complex applications should also be possible. It should be easy to specify multiple complicated queries using user inputs or results from previous queries. Basic constructs, such as branching, looping, and assignment, should be provided for arbitrary processing logics. It should be able to handle various data types in heterogeneous databases. Despite the stateless nature of HTTP, it should be able to support client state management and authentication. It should also make presenting BLOB (Binary Large Object) data easy.
3. Extendibility and easy to learn: The provided constructs should be uniform and easy to learn. It should be easy to add new capabilities without conflicting existing features. The architecture should also be scalable for high performance.
4. Flexibility: The generated application should be able to work with any database, web server, and web browser. It should also be easy to bridge to legacy code.

As shown in Fig. 2, the major components of GWB are the GWB language, its compiler, and system function libraries. To free developers from details of presenting results in HTML texts, GWB provides an HTML-based template language. It adds only several extended tags to bring database access, processing logic, and procedure abstraction into templates. Four system predefined records are provided to easily access program arguments in the templates. GWB also provides a rich set of data types and operators that, coupled with extended tags, makes complicated database application possible. In addition, GWB introduces the virtual session mechanism to maintain client state and perform one-time user authentication.

Functions are essential to express abstract operations. They add extendibility to a language without complicating its syntax. GWB has a rich set of functions to encapsulate high-level operations such as data formatting and input validation. It also

allows user-defined external functions, which are called in the same way as system functions. GWB generates CGI programs that access databases through ODBC libraries, but can be extended to use a three-tier architecture with separate application and database servers without changing its semantics. Because GWB generates ODBC codes and follows CGI standards, it works with any RDBMS, and any web server.

IV. THE GWB LANGUAGE

This section presents the main features of the GWB language and gives an example template. Detailed syntax rules and more examples can be found in [6]. Like most other scripting languages, the syntax of GWB contains two main syntactic categories: expressions and statements. Expressions reside inside statements and are constructed by applying operators on literals, variables, and function calls. Statements are the extended tags that GWB introduces into HTML-based templates. Statements add control and database access operations that are needed to turn an HTML page into an application page. A GWB template thus can be viewed as a base HTML file with GWB statements intermixed.

1. Variables, functions, and expressions

Variables are used to hold user input and intermediary values computed during program execution. Syntactically, a GWB variable is a dollar sign leading identifier, e.g., $customer_id. Three system record variables, $GWB_FORM, $GWB_URL, $GWB_CGI, are provided to access the three sources (HTML form, URL query string, and environment variables) of parameters into a CGI program. Dotted notation is used to qualify individual components in a record, e.g., $GWB_FORM.customer_id denotes an input form field whose name is 'customer_id'. An unqualified name can represent a member of either $GWB_FORM, $GWB_URL, $GWB_CGI, or a template variable. GWB automatically searches $GWB_FORM, $GWB_URL, $GWB_CGI, and user-defined template variables, in that order, for an unqualified name.

GWB is designed to be a compact language with a rich set of system functions for various needs in typical Web/Database applications. A function is a dollar sign leading identifier immediately followed by a left parenthesis, zero or more arguments, and a closing parenthesis, e.g., $GWB_is_date_format ($datefield). Currently, the set of functions includes various HTML formatting, input validation, type coercion, string and list manipulation, session and authentication, etc. GWB also allows user-defined external C functions for hooking up legacy code and for extending GWB's capability.

Table 1. GWB primitive data types

NULL	Error	Boolean	String	Integer	Bigint
Numeric	Float	Date	Time	Timestamp	Blob

In their simplest usage, GWB variables and function calls can be printed to HTML output streams, and incorporated in SQL statements. But GWB offers operators and expressions. Expressions can be composed from variables, literals, functions and fourteen operators. Values of expressions are the sources of processing logic provided by GWB extended tags.

2. Data types

Many template based Web/Database gateway application generators ignore data types and treat all values as character strings. In contrast, GWB provides twelve primitive data types (Table 1) for database and error values, and two compound types, list and record, for basic data structure.

GWB is a dynamically typed language. A variable can hold values of different types throughout its lifetime. Since CGI provides only character string passing between web servers and CGI programs, GWB determines the data type of a variable to be NULL, Integer, Float, String, or List depending on if the value is supplied, the format is appropriate, and multiple values are supplied. For variables corresponding to values from a database, GWB is able to query the database for data types through ODBC API. Most appropriate GWB data types would be assigned for such variables.

GWB has blob type for database BLOB data. BLOB data generally contain images, audio, video, or documents. GWB provides a special function, $GWB_format_blob(var, mime), to automatically save a blob variable's data into a file on the server and generate a proper tag with the SRC attribute invoking a *blob handler* program. This program is passed with the saved file name and the MIME type and is responsible for sending the BLOB data file to clients and removing the data file after being used.

3. Database access statements

Two main statements of GWB are the SQL and RESULTSET statements. The SQL statement brings database connection into a template. It has attributes to specify data source, login id, password, and a query string to be executed. It also has a name, which refers to a record that carries the query execution result. The SQL statement has the following form:

```
<GWB_SQL NAME="sql_name" SOURCE="Data Source" ID
  ="id" PASSWD="passwd" QUERY="SQL query-string">
```

The following example selects from a 'title' table all rows whose 'author' column starts with the prefix specified in variable $author. A record variable, $search_book, would be produced that contains members for query execution status, error, and results

```
<GWB_SQL NAME="earch_book" SOURCE="INF_BOOK_
  DB" ID="guest" PASSWD="public"
            QUERY="select * from title where author like
                  '$author%'">
```

The RESULTSET statement is used to iterate through the result set of a query. The NAME attribute specifies a corresponding GWB SQL statement whose result set is to be iterated through. Since result sets could be very large, attributes are provided to limit the number of rows to retrieve. GWB has a nice mechanism which automatically supplies HTML buttons for scrolling through the result set. Three more attributes are provided to specify the offset between previous and next scroll screens, and the labels on the buttons. The RESULTSET statement has the following form:

```
<GWB_RESULTSET NAME="sql_name" ITERATOR=id
  MAX_COUNT=int_expr START_ROW=int_expr OFFSET=
  int_expr NEXT="labal-text" PREV="label-text">
      {any-statement}*
</GWB_RESULTSET>
```

The following example tests if a SQL statement 'search_book' was executed successfully. It prints out an error message with status code if the SQL was not executed successfully. When successful, the 'author', 'title', and 'year' column values are printed using an iterator named 'b':

```
<GWB_IF ($search_book.status NEQ "OK")>
  Database accessing failure : $search_book.status
<GWB_ELSE>
    <GWB_RESULTSET NAME="search_book" ITERATOR
      ="b">
        Author: $b.author Title: $b.title Publish: $b.year
        <BR>
    </GWB_RESULTSET>
</GWB_IF>
```

4. Miscellaneous statements

Branching and looping constructs are essential in any programming language. GWB provides an IF statement for conditional processing, a FOR statement to iterate through all list elements, and a WHILE statement for general looping. Procedures enable abstraction over a set of statements. The GWB procedure declaration has the following form:

```
<GWB_PROCEDURE p_name ({id {, id}*}){VAR id {, id}*}>
      {any-statement}*
</GWB_PROCEDURE>
```

Parameters are surrounded by parentheses. The VAR list declares variables local to the procedure. Once declared, a procedure can be called as follows:

```
<GWB_CALL p_name({expr {, expr}*})>
```

Any GWB object, including lists and SQL records, can be passed as actual parameters. When called, the statements are evaluated sequentially. Any HTML output generated by these statements are inserted to the HTML stream of the context in which the procedure is invoked.

There are other statements in GWB. <GWB_SET var=expr> assigns the value of an expression to a variable. <GWB_INCLUDE SRC=template_path> is for source code inclusion. <GWB_OUTPUT SRC=html_path> is for inserting an HTML file to output streams at run-time. <GWB_EXEC COMMAND=host_command> is for executing an external host command whose output is inserted into the HTML output stream.

There are also debugging aids in GWB. The function $GWB_print_record($record) prints the contents of the records passed by the user through the web server. The <GWB_SET_DEBUG> statement forces any intermediary output generated by a program to be flushed to the browser. In addition, a stub library is provided so that developers can test out a GWB template without actually doing ODBC connection and executing the SQL statements.

5. COOKIE_HEADER and SET_COOKIE statements

GWB supports client state persistency through the Netscape extension to the HTTP Response Header [6]. This header specifies, among other things, the value and expiration date of a named item, called *cookie*. After receiving a cookie, clients would keep it and determine if the cookie should be sent in HTTP headers based on whether a requested URL matches the cookie's attributes. Cookies can be set using the SET_COOKIE statement:

```
<GWB_SET_COOKIE NAME="quoted_string" VALUE=expr
  EXPIRES=timestamp_expr
DOMAIN=expr PATH=expr>
```

This statement results in immediate output of a

Set-Cookie HTTP response header whose name is the quoted string and the value of the expression. Since response headers precede HTTP main message body, GWB_SET_COOKIE can only appear in GWB_COOKIE_HEADER blocks, which precede all other statements in a GWB template. Within a template, a system record, $GWB_COOKIE, is provided to access all the cookies sent from the client.

6. Authentication and session management

Electronic commerce over the WWW and intranet applications requires both strict authentication and session control. Traditional client-server applications are able to maintain states and manage transactions because they constantly maintain a session with the server. This is impossible, and certainly not efficient, for web applications which use stateless HTTP.

This problem can be addressed by *virtual session*. A virtual session maintains session related information, such as database connections, user privileges, and client states on the servers without passing this information in every http request. A virtual session allows one-time authentication; all subsequent requests would enjoy the same privileges (e.g., id and password) when proved to originate from the same session by verifying a returned session token. An important requirement in supporting virtual session is to avoid session fraud where a session token is deprived of and being used to access sensible data.

Session management is provided through three system functions. $GWB_start_session(session_name, id, minute) creates a session named 'session_name' for user named 'id'. The 'minute' argument specifies after how many minutes this session should expire when there is no further activities associated with the session. After a session is created other templates can call $GWB_verify_session(session_name) to verify a session before further processing. This function is essentially a guard against subsequent processing in a template. $GWB_end_session (session_name) terminates a session.

A session token contains encrypted information including client ip, id, and expiration date. When a session is generated a session token is sent to the client as a cookie and also recorded on the server in a central session file. When accessing other templates which contain a $GWB_verify_session, the client would have to send the session token back to the server, where the template would verify it against client ip, expiration date, and the session file. A successful verification would generate a new session token, which replaces the original one on both the client and the server's session file. This scheme ensures maximum security for session data.

User authentication is provided through two basic authentication functions: $GWB_encrypt(data, key) takes a data string and encrypts it with a key using a DES algorithm; $GWB_authenticate (passwd_file, id, encrypt_pd) opens 'passwd_file' and verifies the encrypted password for 'id' is 'encrypt_pd'.

The following code segment shows a template which authenticates users based on the id and password passed from an HTML form. It starts a session when the authentication is successful. Other pages would verify the session before sending back further page content.

```
... (authentication page starts a session)
<GWB_COOKIE_HEADER>
    <GWB_IF (NOT $GWB_authenticate($passwd_file, $id,
                $GWB_encrypt($passwd, $encrypt_key)))>
        <GWB_SET proceed = "REJECT">
    <GWB_ELSE>
        <GWB_SET proceed = "OK">
        <GWB_SET mysession = $GWB_start_session
         ("mycgi_session", id, 30)>
        ... the session start time recorded
    </GWB_IF>
</GWB_COOKIE_HEADER>
<GWB_IF ($proceed EQ "REJECT)>
        You have not entered the right id and password.
<GWB_ELSE>
        ... successfully authenticated
        ... generate the first screen
        ... session already generated
... (other pages verifying a session)
<GWB_COOKIE_HEADER>
    <GWB_SET proceed = $GWB_verify_session("mycgi_
     session")>
</GWB_COOKIE_HEADER>
<GWB_IF ($proceed EQ "NO_SESSION")>
        Please go to login page to login before using this page.
<GWB_ELSEIF ($proceed EQ "INVALID_SESSION" OR $pro-
 ceed EQ "INVALID_IP")>
        You don't have the right privilege to access this page.
<GWB_ELSEIF ($proceed EQ "EXPIRED_SESSION")>
        Please go to login page to login again.
<GWB_ELSE>
        ... session OK. proceed with the rest of the page.
```

7. An example GWB template

This application is a typical internet database access example. Here the partial name of an electronic product is entered by users from an input form. This template would retrieve matched items from databases. Sounds and pictures of the items would be automatically formulated as anchors and inline images. In case of error a technical person is

automatically notified by running a host command to call a beeper number. This example demonstrates that it is very easy to access databases and present multimedia information in GWB.

```
<html><head>
   <title>Product Information</title>
</head><body>
<GWB_SQL NAME="sel" SOURCE="Items" ID="xxx"
 PASSWD="yyy"
          QUERY="select * from Items where name=
          '%$name%'">
<GWB_IF ($sel.status NEQ "OK")>
     Sorry, database is inaccessible at this moment. Technical
     persons have been notified. Please try again later. <BR>
     <GWB_EXEC COMMAND="beep 9-456-7766">
<GWB_ELSE>
     <GWB_RESULTSET NAME="sel" ITERATOR="i"
      MAX_COUNT=1>
          Product Name: $i.name<BR>
          Price: $i.price<BR>
          <GWB_IF ($i.pic NEQ GWB_undefined)>
               Picture: $GWB_format_blob($i.pic, "im-
               age/gif")<BR>
          </GWB_IF>
          <GWB_IF ($i.music NEQ GWB_undefined)>
               Music: $GWB_format_blob($i.music,
               "audio/wav")<BR>
          </GWB_IF>
     </GWB_RESULTSET>
</GWB_IF>
</body></html>
```

V. DISCUSSION AND FUTURE WORK

GWB is a simple yet powerful database application generator for the World Wide Web. Its programming model is HTML-based. The GWB language adds data types, operators, expressions, functions, and extended tags into HTML for application processing logics and database access. Any number of SQL statements and arbitrarily complex processing logics can be programmed within a template. The procedure modularization, authentication, and session management utilities make it suitable for large-scale intranet applications.

The architecture of GWB generated applications is CGI. The advantage of using CGI is that it is compatible with any Web server. With CGI architecture, extending the functionality can also be easily achieved by adding more system functions and user-defined external functions.

Currently GWB is being transformed into a product by Trilogy Technology International, Inc. [21]. The implementation on Microsoft Windows NT and SunOS/Solaris have been completed. Ports to other Unix platforms are under construction. Several internet/intranet applications have been developed and deployed using GWB [8]. Throughout the process, we have found the following features of GWB very useful for constructing Web/RDBMS applications.

1. The same templates can be used to generate applications for both Windows NT and Unix platforms.
2. Legacy code can be integrated into GWB templates by linking existing C code or executing a shell command at run-time.
3. Navigation through a result set is supported with previous and next buttons that the system generates automatically.
4. User-defined procedure support abstractions over common behaviors and modularization of GWB templates.
5. Automatically fetches, links, and cleans up database BLOBs.
6. The rich set of data types and the associated formatting functions make the processing of number, date, time, and timestamp data from the database easy and flexible.
7. Built-in support of user authentication and client state management.

GWB can still be improved in many ways. On the programming side, an integrated development environment is being planned. A language mechanism for performing file I/O is highly desirable. The major necessary improvements lie on the architecture side. Basically, there are two concerns. The first is about scalability--how to reduce the overhead of forking CGI programs and enhance database access performance? The second relates to inconvenience and overhead of passing client state as HTTP cookies.

To address the scalability issue the next generation of GWB could adopt a three-tier architecture, such as in Fig. 3, to distribute the requesting load. One possible approach would configure GWB application servers as server-side Java applications. The GWB template source could be compiled into Java classes, which are dynamically loaded and run by application servers.

In addition, the protocol between application and database servers should be redesigned to facilitate caching database connections and cursor states within database servers. Current implementation of GWB terminates database connections and closes all open SQL statements when a CGI exits. When navigating through a large result set through windows (e.g., using GWB_RESULTSET statement) in multiple invocation of a CGI, the same SELECT statement has to be re-executed again in each CGI invocation. In addition, transactions cannot cross multiple CGI's, again, due to the lack of persistent database connections.

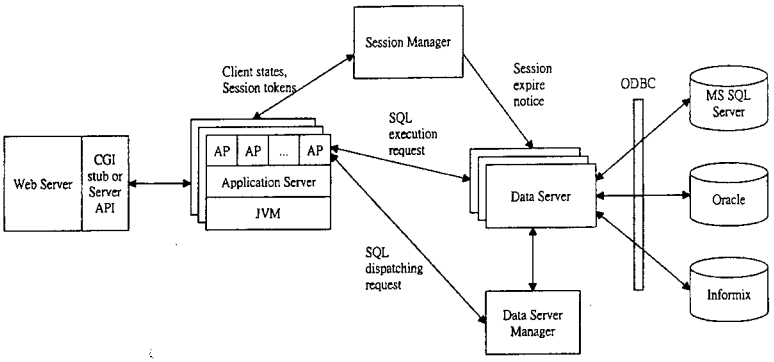


Fig. 3. GWB Three-tier Architecture.

Our approach is to design a persistent data server which executes SQL statements on behalf of CGI's. Since the data server is persistent, both database connections and open SQL statements can stay open after a CGI exits. It is thus possible to provide cross-CGI transactions and cursor state preservation. This scheme also opens the possibility of all CGI's sharing database connections. It has the advantage of further enhancing database access performance since connection set up is time-consuming.

Currently session management is provided by GWB utility functions. It is limited in that session data have to be passed between client and server. A *session manager* is being added to the next generation of GWB. It would perform the job of generating session tokens and keeping arbitrary client data on the server. Since the data stays on the server it can be of any data type and size. Server-side client state persistence is thus more flexible than passing the client state as HTTP cookies.

REFERENCES

1. Allaire, Inc., *Cold Fusion 2.0 User's Guide*, November 1996. Available at <URL: http://www.allaire.com/products/coldfusion/20/userguide/index.htm>
2. Berners-Lee, T., R. Cailiau, A. Luotonen, H.F. Nielsen, and A. Secret. "The World Wide Web," *Communication of the ACM*, Vol. 37, No. 8, pp. 76-82, (1994).
3. Chen, K. and W.J. Lin, *GWB User Manual*. Available at <URL: http:://www.cs.ntust.edu.tw/~chenk/gwb/>
4. Duan, N.N., "Distributed Database Access in a Corporate Environment Using Java," Proc. of Fifth International World Wide Web Conference, Paris, France (1996). Available at <URL: http://www5conf.inria.fr/fich_html/papers/P23/Overview. html>
5. Hunter, A., R.I. Ferguson, and S. Hedges. "SWOOP: An Application Generator for ORACLE/WWW Systems," *The World Wide Web Journal*, Vol. 1, No. 1, 1996. Available at <URL: http:// www.w3j.com/1/hunter.207/paper/207.html >
6. Kristol, D.M. and L. Montulli, "HTTP State Management Mechanism," *The World Wide Web Journal*, Vol. 1, No. 4, Fall 1996. <URL: http://www.w3j.com/4/s3.kristol.html >
7. Letovsky, S.I., Johns Hopkins Medical Institutes. *What Web/Genera Does*, December 8, 1995. Available at <URL: http://gdbdoc.gdb.org/letovsky/genera/genexamples.html>
8. Lin, T., W.J. Lin, and K. Chen. *Developing Large-scale Web-based Applications: An Experience Report*, In preparation.
9. National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign. *The Common Gateway Interface*. Available at <URL: http://hoohoo.ncsa.uiuc.edu/cgi/>
10. Naumann, M., European Southern Observatory. *WDB - A Web interface to SQL Databases*, July 25, 1996. Available at <URL: http://archive.eso.org/wdb/html/>
11. NetDynamics, Inc. *NetDynamics 4.0: Enterprise Network Application Platform*. Available at <URL: http://www.netdynamics.com/products/whitepaper/nd40techbrief.pdf>
12. Netscape Communications Corporation. *The Netscape Server API*. Available at <URL: http://home.netscape.com/newsref/std/server_api.html>
13. Nguyen, T. and V. Srinivasan, "Accessing Relational Databases from the World Wide Web," Proc. of 1996 ACM SIGMOD, Montreal, Quebec, pp. 529-540.
14. Open Market, Inc. *FastCGI: A High-Performance Web Server Interface*, April 1996. Available at <URL: http://www.fastcgi.com/kit/doc/fastcgi-whitepaper/fastcgi.htm>
15. *Perl and Oracle*. April 10, 1996. Available at

<http://www.bf.rmit.edu.au/~orafaq/perlish.html>
16. Richmond, A., *Web Developer's Virtual Library: Database*. Available at <URL: http://Stars.com/Vlib/Software/Database.html>
17. Rosenblatt, B., "Sybase's Web tools strategy," *SunWorld Online*, March 1996. Available at <URL: http://www.sun.com/sunworldonline/swol-03-1996/swol-03-cs.html>
18. Rowe, J., *Accessing a Database Server via the World Wide Web-Existing Web/Database Gateway Products for All Platforms*. <URL: http://cscsun1.larc.nasa.gov/~beowulf/db/all_products.html>
19. StormCloud Development Corporation. *WebDBC Quick Reference*, November 1996. Available at <http://www.ndev.com/wdbc3/default.htm>
20. Sybase, Inc., *sybperl*. Available at <URL: http://www.sybase.com/products/internet/websql/index.html>
21. Trilogy Technology International, Inc. *OpenPath Web*. Available at <http://www.openpath.com>
22. Weblogic, Inc., *Database connectivity in the multitier framework*. Available at <URL: http://www.weblogic.com/whitepapers/t3db.html>
23. World Wide Web Consortium. *HyperText Markup Language (HTML)*. Available at <URL: http://www.w3.org/pub/WWW/MarkUp/>
24. World Wide Web Consortium. *HTTP - Hypertext Transfer Protocol*. Available at <URL: http://www.w3.org/pub/WWW/Protocols/>

APPENDIX A. GWB EXTENSION TAGS

<GWB_COKIE_HEADER> </GWB_COKIE_HEADER>
<GWB_PROCEDURE> </GWB_PROCEDURE>
<GWB_CALL>
<GWB_SET>
<GWB_SET_COOKIE>
<GWB_IF> <GWB_ELSEIF> <GWB_ELSE> </GWB_IF>
<GWB_FOR> </GWB_FOR>
<GWB_WHILE> </GWB_WHILE>
<GWB_SQL>
<GWB_RESULTSET>
<GWB_EXEC>
<GWB_OUTPUT>
<GWB_INCLUDE>
<GWB_SET_DEBUG>

Discussions of this paper may appear in the discussion section of a future issue. All discussions should be submitted to the Editor-in-Chief.

Manuscript Received: May 14, 1997
Revision Received: Sep. 26, 1997
and Accepted: Mar. 07, 1998

全球資訊網資料庫應用程式產生器

林維志

國立政治大學資訊科學系

陳 恭

國立台灣科技大學資訊管理系

摘 要

近年來全球資訊網發展迅速，已從一單純之超媒體文件網擴大爲一主從應用程式之新平台。連結全球資訊網與資料庫間之閘道應用程式即爲其中之重要範例。本文描述一名之爲GWB的全球資訊網資料庫應用程式產生器，爲了幫助程式設計師快速發展此類程式，GWB 以超文件標記語言（HTML）爲基礎，外加一些存取資料庫及控制流程的標記的高階應用語言，讓程式設計師將所要做的資料庫存取運算與所得結果安排直接以類似超文件標記語言的方式表達出來，不必爲些繁瑣易錯的細節所煩，也不需應付龐大的資料庫應用程式介面，只需專注於發展應用程式所需之邏輯與運算。GWB將以此高階應用語言寫成之模板編譯成符合ODBC介面的資料庫應用程式，大幅縮短程式的開發時程。

關鍵字：全球資訊網，通用閘道介面，資料庫。

RANDOM VIBRATION OF MULTI-SPAN MINDLIN PLATE DUE TO MOVING LOAD

Rong-Tyai Wang* and Tsang-Yuan Lin
Department of Engineering Science,
National Cheng Kung University
Tainan, Taiwan 701 ROC.

Key Words: Multi-span, Mindlin plate, moving load, mean value, variance.

ABSTRACT

In this paper, the method of modal analysis is presented to study the random vibration of multi-span Mindlin plates due to a load moving at a constant velocity. The moving load is considered to be a stationary process with a constant mean value and a variance. Four types of variances are considered in this study: white noise, exponential, exponential cosine, and cosine. The effect of both velocity and statistical characteristics of the load, and the effect of the span number of the multi-span plate on the mean value, variance of deflection and moment of the structure are investigated. The results of the multi-span Mindlin plate are compared with those of a multi-span classical plate.

I. INTRODUCTION

Plate vibration has been studied for more than a century [8]. In studying the problem of load moving on a plate, both magnitude and velocity of the load are generally considered to be constants [1, 10]. In real situations, both magnitude and velocity of a moving load, such as traffic loads on a road, wind loads to a tall building, turbulence to wings, etc., cannot be described deterministically. The vibration of structures due to these kinds of load is unpredictable. Even worse, they may cause a disaster.

The random vibration of simple structures due to a distributed stationary excitation has been studied for more than thirty years [3, 4]. In recent years, progress in technology makes the weight and velocity of vehicles more complex than ever before. The characteristics of moving loads can only be estimated by a stochastic process. The problem of random vibration induced by loads moving on structures is different from the previous cases. Many studies about random vibration of a simple beam due to moving loads have been reported [5-7, 11, 12]. Relative to beams, plates are also widely used in construction, e.g., decks of a bridge. Plate structures are usually built of many similar units to reduce cost and simplify the process of construction. The maximum deflection and the maximum moment of a multi-span plate induced by a moving load are always greater than those of a plate induced by the same static load [13]. Furthermore, there is a critical velocity at which the multi-span plate deforms significantly. However, the influence on responses of a multi-span plate by random loads traveling on the structure has never been investigated. The responses of a multi-span plate depend on the velocity, time, mean value and variance of random loads. The random vibration of a multi-span plate due to moving loads will induce fatigue in the structure. Therefore, the random vibration of a multi-span plate due to moving loads is an important problem in structural dynamics.

Classical plate theory leads to erroneous results

*Correspondence addressee

for both a large ratio of thickness to length (or width) and high modes. The Mindlin plate theory [9] is, therefore, considered in this paper to study the vibration of a multi-span plate due to a random load moving on the structure. The multi-span plate is considered to be homogeneous and isotropic with Young's modulus E, shear modulus G, cross-sectional area A, density ρ, second moment of area I and shear coefficient κ. Furthermore, each span has equal width b, thickness h and length a. To obtain the basic phenomena of responses of the multi-span plate, the random load is traveling at a constant velocity. The random load has a constant mean value and a deviation. Four types of variances of the deviation are considered. They are white noise process, exponential process, exponential cosine process and cosine process. In addition, the effect of both velocity and variance type of the load, and the effect of span number of the multi-span plate on both the mean value and the variance of response are investigated. Results obtained for the multi-span Mindlin plate are compared with those of a multi-span classical plate.

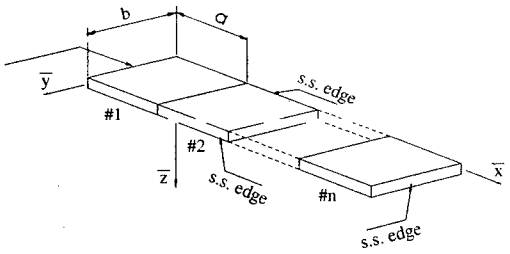


Fig. 1. A periodical simply supported Mindlin plate.

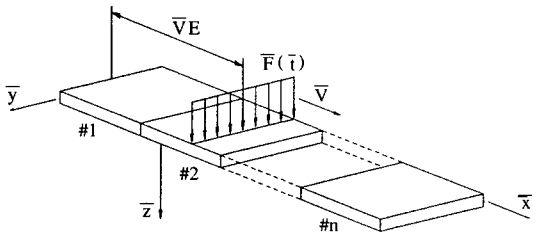


Fig. 2. A random load moves on a multi-span Mindlin plate.

II. GOVERNING EQUATIONS

A Mindlin plate on periodically simple supports is depicted in Fig. 1. The responses $\bar{W}$, $\bar{\Psi}_{\bar{x}}$, $\bar{\Psi}_{\bar{y}}$, $\bar{m}_{\bar{x}}$, $\bar{m}_{\bar{y}}$, $\bar{m}_{\bar{x}\bar{y}}$, $\bar{q}_{\bar{x}}$ and $\bar{q}_{\bar{y}}$ of the multi-span plate due to the distributed load $\bar{P}(\bar{x},\bar{y},\bar{t})$ can be expressed in terms of the superposition of mode shape functions $\bar{W}^{(ij)}$, $\bar{\Psi}_{\bar{x}}^{(ij)}$, $\bar{\Psi}_{\bar{y}}^{(ij)}$, $\bar{M}_{\bar{x}}^{(ij)}$, $\bar{M}_{\bar{y}}^{(ij)}$, $\bar{M}_{\bar{x}\bar{y}}^{(ij)}$, $\bar{Q}_{\bar{x}}^{(ij)}$ and $\bar{Q}_{\bar{y}}^{(ij)}$, respectively, as the forms

$$(\bar{W},\bar{\psi}_{\bar{x}},\bar{\psi}_{\bar{y}})(\bar{x},\bar{y},\bar{t}) = \sum_{j=1}\sum_{i=1} A_{ij}(\bar{t})(\bar{W}^{(ij)},\bar{\Psi}_{\bar{x}}^{(ij)},\bar{\Psi}_{\bar{y}}^{(ij)})(\bar{x},\bar{y}), \tag{1a}$$

$$(\bar{m}_{\bar{x}},\bar{m}_{\bar{y}},\bar{m}_{\bar{x}\bar{y}})(\bar{x},\bar{y},\bar{t}) = \sum_{j=1}\sum_{i=1} A_{ij}(\bar{t})(\bar{M}_{\bar{x}}^{(ij)},\bar{M}_{\bar{y}}^{(ij)},\bar{M}_{\bar{x}\bar{y}}^{(ij)})(\bar{x},\bar{y}), \tag{1b}$$

$$(\bar{q}_{\bar{x}},\bar{q}_{\bar{y}})(\bar{x},\bar{y},\bar{t}) = \sum_{j=1}\sum_{i=1} A_{ij}(\bar{t})(\bar{Q}_{\bar{x}}^{(ij)},\bar{Q}_{\bar{y}}^{(ij)})(\bar{x},\bar{y}), \tag{1c}$$

in which the ijth modal amplitude A_{ij} is obtained by solving the equation [13]

$$\frac{d^2A_{ij}}{d\bar{t}^2} + \bar{\omega}_{ij}^2 A_{ij} = g_{ij}(\bar{t}). \tag{2}$$

In Eq. (2) $\bar{\omega}_{ij}$ is the ijth modal frequency and the corresponding excitation $g_{ij}(\bar{t})$ is

$$g_{ij}(\bar{t}) = \int_0^{na}\int_0^b \bar{P}(\bar{x},\bar{y},\bar{t})\bar{W}^{(ij)}(\bar{x},\bar{y})\,d\bar{x}\,d\bar{y}/s_{ij}, \tag{3}$$

where the ijth modal mass S_{ij} is

$$s_{ij} = \int_0^{na}\int_0^b eh\left(\frac{h^2}{12}\bar{\Psi}_{\bar{x}}^{(ij)^2} + \frac{h^2}{12}\bar{\Psi}_{\bar{y}}^{(ij)^2} + \bar{W}^{(ij)^2}\right)d\bar{x}\,d\bar{y}. \tag{4}$$

The initial conditions of the plate are set at zeros. The response history of the ijth modal amplitude $A_{ij}(\bar{t})$ is

$$A_{ij}(\bar{t}) = \int_{\infty}^{-\infty} u_{ij}(\bar{\tau})g_{ij}(\bar{t}-\bar{\tau})\,d\bar{\tau}, \tag{5}$$

in which the impulse response $u_{ij}(\bar{t})$ is

$$u_{ij}(\bar{t}) = \begin{cases} \sin(\bar{\omega}_{ij}\bar{t})/\bar{\omega}_{ij} & (0<\bar{t}) \\ 0 & (\bar{t}\le 0). \end{cases} \tag{6}$$

III. MOTION OF A RANDOM LOAD ON THE PLATE

A load, which is uniformly distributed along the $\bar{y}$-axis, moving on the plate at the constant velocity $\bar{v}$ in the direction of the $\bar{x}$-axis is depicted in Fig. 2. The expression of the load is

$$\bar{P}(\bar{x},\bar{y},\bar{t}) = \bar{F}(\bar{t})U(\bar{y})\delta(\bar{x}-\bar{v}\,\bar{t}), \tag{7}$$

where $U(\bar{y})$ is the unit step function. The magnitude $\bar{F}(\bar{t})$ of the load is considered to be a random process with a mean value $\langle\bar{F}(\bar{t})\rangle$ and a centered random value $\bar{f}(\bar{t})$. The covarianc between $\bar{f}(\bar{t}_1)$ and $\bar{f}(\bar{t}_2)$ is denoted as $C_{\bar{f}}(\bar{t}_1,\bar{t}_2)$, i.e.,

$$C_{\bar{f}}(\bar{t}_1,\bar{t}_2)=\left\langle \bar{f}(\bar{t}_1)\ \bar{f}(\bar{t}_2)\right\rangle . \tag{8}$$

The notation $\langle\ \rangle$ is the mean value operator. The respective histories of the ijth modal excitation $g_{ij}(\bar{t})$ and its corresponding mean value $\langle g_{ij}(\bar{t})\rangle$, and the covariance $C_{g_{ij}g_{kl}}(\bar{t}_1,\bar{t}_2)$ between $g_{ij}(\bar{t}_1)$ and $g_{kl}(\bar{t}_2)$ are

1) $0\le \bar{t}\ ,\ \bar{t}_1,\ \bar{t}_2 \le na/\bar{v}$

$$g_{ij}(\bar{t})=\bar{F}(\bar{t})\int_0^b \bar{W}^{(ij)}(\bar{v}\,\bar{t},\bar{y})\,d\bar{y}/s_{ij}, \tag{9a}$$

$$\left\langle g_{ij}(\bar{t})\right\rangle=\left\langle \bar{F}(\bar{t})\right\rangle\int_0^b \bar{W}^{(ij)}(\bar{v}\,\bar{t},\bar{y})\,d\bar{y}/s_{ij}, \tag{9b}$$

$$C_{g_{ij}g_{kl}}(\bar{t}_1,\bar{t}_2)$$

$$=C_{\bar{f}\bar{f}}(\bar{t}_1,\bar{t}_2)\int_0^b \bar{W}^{(ij)}(\bar{v}\,\bar{t}_1,\bar{y})\,d\bar{y}\int_0^b \bar{W}^{(kl)}(\bar{v}\,\bar{t}_2,\bar{y})$$

$$\cdot d\bar{y}/s_{ij}s_{kl} \tag{9c}$$

2) $na/\bar{v}<\bar{t}\ ,\ \bar{t}_1$ (or $\bar{t}_2$)

$$g_{ij}(\bar{t})=0,\quad \left\langle g_{ij}(\bar{t})\right\rangle=0,\quad C_{g_{ij}g_{kl}}(\bar{t}_1,\bar{t}_2)=0. \tag{10a, b, c}$$

The mean value histories of the ijth modal amplitude A_{ij}, transverse deflection $\bar{w}$, and moment $\bar{m}_{\bar{x}}$, respectively, are

$$\left\langle A_{ij}(\bar{t})\right\rangle=\int_{-\infty}^{\infty} u_{ij}(\bar{t}-\bar{\tau})\left\langle g_{ij}(\bar{\tau})\right\rangle d\bar{\tau},$$

$$\left\langle \bar{w}(\bar{x},\bar{y},\bar{t})\right\rangle=\sum_{i=1}\sum_{j=1}\bar{W}^{(ij)}(\bar{x},\bar{y})\left\langle A_{ij}(\bar{t})\right\rangle, \tag{11a, b}$$

$$\left\langle \bar{m}_{\bar{x}}(\bar{x},\bar{y},\bar{t})\right\rangle=\sum_{i=1}\sum_{j=1}\bar{M}_{\bar{x}}^{(ij)}(\bar{x},\bar{y})\left\langle A_{ij}(\bar{t})\right\rangle. \tag{11c}$$

The covariance $C_{A_{ij}A_{kl}}(\bar{t}_1,\bar{t}_2)$ between $A_{ij}(\bar{t}_1)$ and $A_{kl}(\bar{t}_2)$ is

$$C_{A_{ij}A_{kl}}(\bar{t}_1,\bar{t}_2)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} u_{ij}(\bar{t}_1-\bar{\tau}_1)u_{kl}(\bar{t}_2-\bar{\tau}_2)C_{g_{ij}g_{kl}}(\bar{\tau}_1-\bar{\tau}_2)$$

$$\cdot d\bar{\tau}_1 d\bar{\tau}_2. \tag{12}$$

The deflection covariance $C_{\bar{w}}(\bar{x}_1,\bar{x}_2,\bar{y}_1,\bar{y}_2,\bar{t}_1,\bar{t}_2)$ between $\bar{w}(\bar{x}_1,\bar{y}_1,\bar{t}_1)$ and $\bar{w}(\bar{x}_2,\bar{y}_2,\bar{t}_2)$ is

$$C_{\bar{w}}(\bar{x}_1,\bar{x}_2,\bar{y}_1,\bar{y}_2,\bar{t}_1,\bar{t}_2)$$

$$=\sum_{i=1}\sum_{j=1}\sum_{k=1}\sum_{l=1}\bar{W}^{(ij)}(\bar{x}_1,\bar{y}_1)\bar{W}^{(kl)}(\bar{x}_2,\bar{y}_2)C_{A_{ij}A_{kl}}(\bar{t}_1,\bar{t}_2). \tag{13a}$$

The moment covariance $C_{\bar{m}_{\bar{x}}}(\bar{x}_1,\bar{x}_2,\bar{y}_1,\bar{y}_2,\bar{t}_1,\bar{t}_2)$ between $\bar{m}_{\bar{x}}(\bar{x}_1,\bar{y}_1,\bar{t}_1)$ and $\bar{m}_{\bar{x}}(\bar{x}_2,\bar{y}_2,t_2)$ is

$$C_{\bar{m}_{\bar{x}}}(\bar{x}_1,\bar{x}_2,\bar{y}_1,\bar{y}_2,\bar{t}_1,\bar{t}_2)$$

$$=\sum_{i=1}\sum_{j=1}\sum_{k=1}\sum_{l=1}\bar{M}_{\bar{x}}^{(ij)}(\bar{x}_1,\bar{y}_1)M_{\bar{x}}^{(kl)}(\bar{x}_2,\bar{y}_2)C_{A_{ij}A_{kl}}(\bar{t}_1,\bar{t}_2). \tag{13b}$$

Since $\bar{f}(\bar{t})$ is a stationary process, i.e.,

$$C_{\bar{f}}(\bar{t}_1,\bar{t}_2)=C_f(\bar{t}_1-\bar{t}_2), \tag{14}$$

the covariance $C_{g_{ij}g_{kl}}(\bar{t}_1,\bar{t}_2)$ and the covariance $C_{A_{ij}A_{kl}}(\bar{t}_1,\bar{t}_2)$, respectively, are

$$C_{g_{ij}g_{kl}}(\bar{t}_1,\bar{t}_2)$$

$$=C_{\bar{f}}(\bar{t}_1-\bar{t}_2)\int_0^b \bar{W}^{(ij)}(\bar{v}\,\bar{t}_1,\bar{y})\,d\bar{y}\int_0^b \bar{W}^{(kl)}(\bar{v}\,\bar{t}_2,\bar{y})$$

$$\cdot d\bar{y}/(s_{ij}s_{kl}),\qquad 0\le\bar{t}_1,\ \bar{t}_2\le na/\bar{v}, \tag{15a}$$

or

$$C_{g_{ij}g_{kl}}(\bar{t}_1,\bar{t}_2)=0,\quad na/\bar{v}\le\bar{t}_1,\ \bar{t}_2, \tag{15b}$$

$$C_{A_{ij}A_{kl}}(\bar{t}_1,\bar{t}_2)$$

$$=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}u_{ij}(\bar{t}_1-\bar{\tau}_1)u_{kl}(\bar{t}_2-\bar{\tau}_2)C_{g_{ij}g_{kl}}(\bar{\tau}_1,\bar{\tau}_2)$$

$$\cdot d\bar{\tau}_1 d\bar{\tau}_2. \tag{16}$$

Furthermore, the variances of deflection $\bar{w}$ and moment $\bar{m}_{\bar{x}}$ of the plate are denoted, respectively, as $\bar{\sigma}_{\bar{x}}^2(\bar{x},\bar{y},\bar{t})$ and $\bar{\sigma}_{\bar{m}_{\bar{x}}}^2(\bar{x},\bar{y},\bar{t})$, which are

$$\bar{\sigma}_{\bar{w}}^2(\bar{x},\bar{y},\bar{t})=C_{\bar{w}}(\bar{x},\bar{x},\bar{y},\bar{y},\bar{t},\bar{t})$$

$$=\sum_{i=1}\sum_{j=1}\sum_{k=1}\sum_{l=1}\bar{W}^{(ij)}(\bar{x},\bar{y})\bar{W}^{(kl)}(\bar{x},\bar{y})C_{A_{ij}A_{kl}}(\bar{t},\bar{t}), \tag{17a}$$

$$\bar{\sigma}_{\bar{m}_{\bar{x}}}^2(\bar{x},\bar{y},\bar{t})=C_{\bar{m}_{\bar{x}}}(\bar{x},\bar{x},\bar{y},\bar{y},\bar{t},\bar{t})$$

$$=\sum_{i=1}\sum_{j=1}\sum_{k=1}\sum_{l=1}\bar{M}_{\bar{x}}^{(ij)}(\bar{x},\bar{y})M_{\bar{x}}^{(kl)}(\bar{x},\bar{y})C_{A_{ij}A_{kl}}(\bar{t},\bar{t}). \tag{17b}$$

The following four types of covariances are considered (see Figs. 3(a)~3(d)) in the study:
(1) white noise

$$C_{\bar{f}}(\bar{\tau})=\bar{\sigma}_0^2\delta(\bar{\tau}) \tag{18a}$$

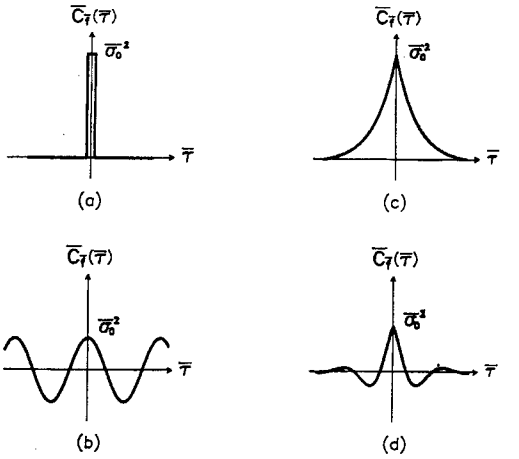


Fig. 3. Four types of variances of the load: (a) white noise, (b) cosine, (c) exponential and (d) exponential cosine

(2) cosine wave

$$C_{\bar{f}}(\bar{\tau})=\bar{\sigma}_0^2\cos(\bar{\omega}_0\bar{\tau}) \tag{18b}$$

(3) exponential

$$C_{\bar{f}}(\bar{\tau})=\bar{\sigma}_0^2 e^{-\bar{\omega}_g\bar{\tau}} \tag{18c}$$

(4) exponential cosine

$$C_{\bar{f}}(\bar{\tau})=\bar{\sigma}_0^2 e^{-\bar{\omega}_g\bar{\tau}}\cos(\bar{\omega}_0\bar{\tau}) \tag{18d}$$

IV. ILLUSTRATIVE EXAMPLES AND DISCUSSION

To illustrate the numerical results in this study, the non-dimensional variables are introduced as

$$x=\bar{x}/a,\ y=\bar{y}/a,\ w=\bar{w}/h,$$

$$(\psi_x,\psi_y)=(\bar{\psi}_{\bar{x}}a,\bar{\psi}_{\bar{x}}b)/h,\ \lambda=a/b,\ t=(D/\rho ha^4)^{1/2}\bar{t},$$

$$(q_x,q_y)=(\bar{q}_{\bar{x}}a,\bar{q}_{\bar{x}}b)/\kappa Gh^2,\ r=h/a,$$

$$(m_x,m_y,m_{xy})=(\bar{m}_{\bar{x}}a^2,\ \bar{m}_{\bar{x}}b^2,\ \bar{m}_{\bar{x}\bar{y}}ab)/Dh,$$

$$F(x,y,t)=\bar{F}(\bar{x},\bar{y},\bar{t})a^4/Dh,\ \omega=(D/\rho ha^4)^{-1/2}\bar{\omega},$$

$$(\bar{F}_0,\bar{\alpha}_0)=(F_0,\alpha_0)L^4/EI\eta,\ \text{and}\ \alpha_M=\bar{v}(\rho/E)^{1/2}$$

in which α_M is the velocity ratio. Moreover, the data μ=0.3, κ=0.85 [2], λ=1 , r=0.1 and $\langle\bar{F}(\bar{t})\rangle$=1 are taken for the purpose of numerical analysis. The responses along the line y=0.5 of the plate are investigated as well. It is known that the mode shape functions of i (or k)=1~10 and j (or l)=1~20 of the plate are sufficient to be employed in the method of modal analysis in the numerical computation [13]. The velocity range considered in this section is $0\le\alpha_M\le 0.28$. The following parameters are defined to illustrate the

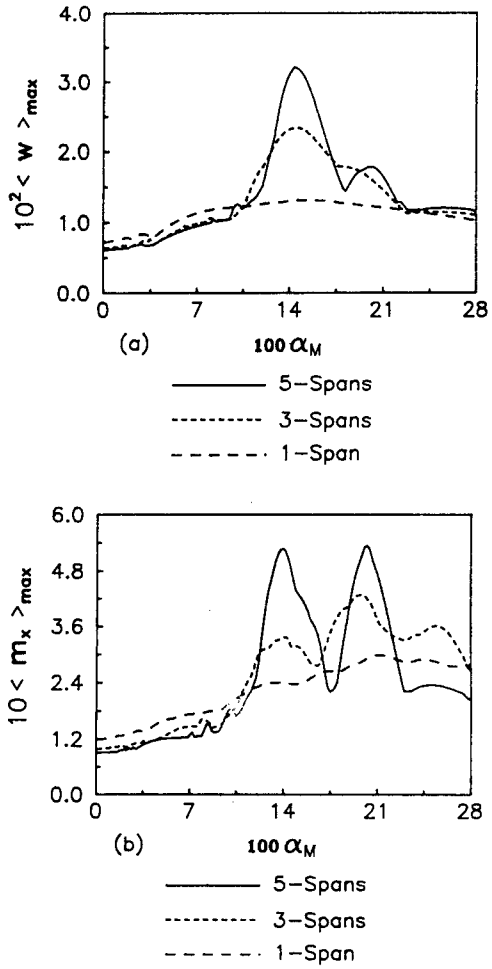


Fig. 4. Span number effect on (a) the $\langle w\rangle_{max}$–α_M distribution and (b) the $\langle m_x\rangle_{max}$–α_M distribution of a multi-span Mindlin plate.

numerical results: maximum $\langle w\rangle$ during the motion of the load, $\langle w\rangle_{max}$; maximum $\langle m_x\rangle$ during the motion of the load, $\langle m_x\rangle_{max}$; position of $\langle w\rangle_{max}$ during the motion of the load, $X_{<W>}$; position of $\langle m_x\rangle_{max}$ during the motion of the load, $X_{<m_x>}$; maximum variance of w during the motion of the load, $\sigma^2_{w,max}$; maximum variance of m_x during the motion of the load, $\sigma^2_{m_x,max}$; position of $\sigma^2_{w,max}$ during the motion of the load, $(X_\sigma)_w$; position of $\sigma^2_{m_x,max}$ during the motion of the load, $(X_\sigma)_{m_x}$; velocity ratio at which $\langle w\rangle_{max}$ appears, α_C; velocity ratio at which $\sigma_{w,max}$ appears, α_σ;

1. Mean value

The effects of span number on the $\langle w\rangle_{max}$–α_M distribution and the $\langle m_x\rangle_{max}$–α_M distribution of a multi-span Mindlin plate are shown in Figs. 4(a) and 4(b), respectively. The higher span number implies

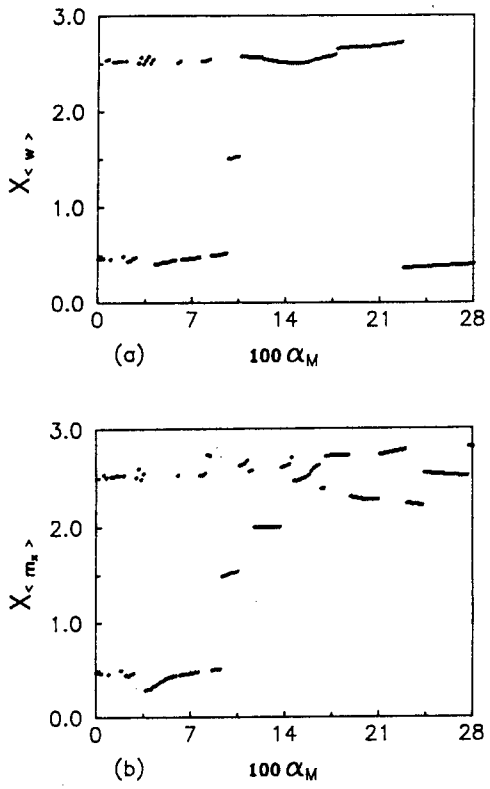


Fig. 5. (a) The $X_{\langle w\rangle}-\alpha_M$ distribution and (b) the $X_{\langle m_x\rangle}-\alpha_M$ distribution of a three-span Mindlin plate.

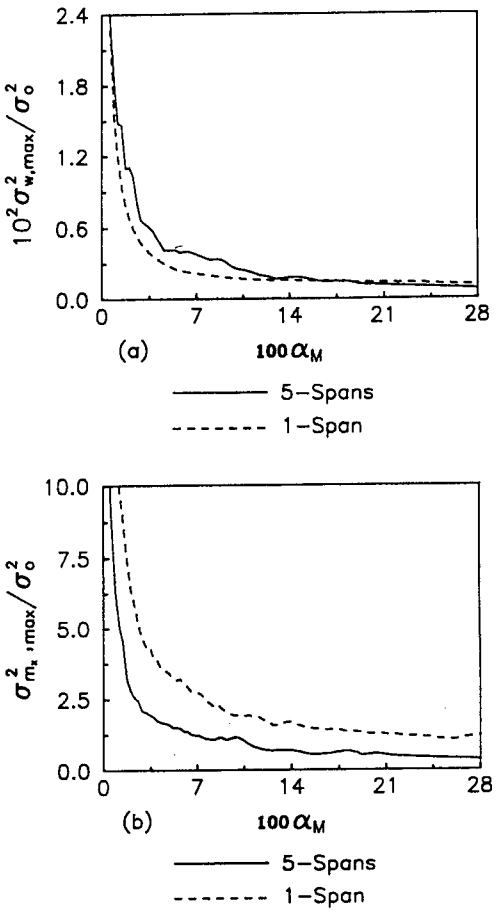


Fig. 6. Span number effect on (a) the $\sigma_{w,\max}-\alpha_M$ distribution and (b) the $\sigma_{w_x,\max}-\alpha_M$ distribution of a multi-span Mindlin plate due to a white noise process.

a heavier plate. The load can be regarded as a quasi-static load within the low velocity range $0\le\alpha_M\le0.1$. Therefore, as the span number increases, both $\langle w\rangle_{\max}$ and $\langle m_x\rangle_{\max}$ of the multi-span Mindlin plate decrease. The effect of a bending wave on the vibration of plate is more apparent for a higher span number and a higher velocity. As a result, both figures illustrate that α_c and both absolute $\langle w\rangle_{\max}$ and $\langle m_x\rangle_{\max}$ increase as the span number increases. Furthermore, α_c is more apparent for the higher span number.

The $X_{\langle w\rangle}-\alpha_M$ distribution and the $X_{\langle m_x\rangle}-\alpha_M$ distribution of a three-span Mindlin plate are displayed in Figs. 5(a) and 5(b), respectively. No reaction moment occurs along either the first simply supported edge (x=0) or the fourth simply supported edge (x=3). Therefore, $<w>_{\max}$ always appears near the mid-point of either the first span or the third span. Within a low velocity range $0\le\alpha_M\le0.07$, $\langle m_x\rangle_{\max}$ occurs during the load moving on the plate. Therefore, $\langle m_x\rangle_{\max}$ appears approximately near the mid-point of either the first span or the third span within the velocity range. For a load moving at a supercritical velocity, $\langle m_x\rangle_{\max}$ will appear after the load has left the plate. Therefore, $\langle m_x\rangle_{\max}$ occurs within the third span for a load traveling at a supercritical velocity.

2. White noise

The frequency range of the power spectrum of the white noise process extends from negative infinity to positive infinity. Therefore, all modal responses of the Mindlin plate are induced by the white noise process. The slow moving load results in a long duration of forced vibration of the plate. Therefore, the plate will be in the steady state of vibration as the duration of forced vibration goes to infinite, i.e., the velocity of the load approaches zero. Under this circumstance, the plate will be in resonance. Therefore, both $\sigma^2_{w,\max}$ and $\sigma^2_{m_x,\max}$ will be infinite as α_M approaches zero. Moreover, both $\sigma^2_{w,\max}$ and $\sigma^2_{m_x,\max}$ rapidly decrease as α_M increases. The higher span number causes the plate to have a longer duration of forced vibration. Therefore, in Fig. 6(a) it is

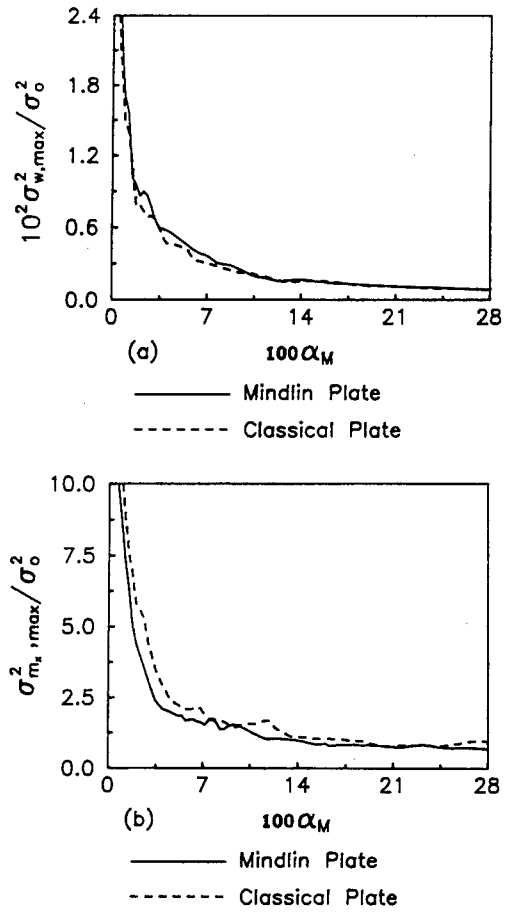


Fig. 7. Effects of two different plate theories on (a) the $\sigma_{w,\max}-\alpha_M$ distribution and (b) the $\sigma_{m_x,\max}-\alpha_M$ distribution of a three-span plate due to a white noise process.

shown that the larger span number implies a larger $\sigma^2_{w,\max}$. However, a higher span number means a higher number of simply supported edges. Therefore, a higher span number causes a smaller $\sigma^2_{m_x,\max}$ due to the heavier mass of the plate and the higher number of reaction moment, as indicated in Fig. 6(b).

The effects of two different plate theories on the $\sigma^2_{w,\max}-\alpha_M$ distribution and the $\sigma^2_{m_x,\max}-\alpha_M$ distribution of a three-span plate are shown in Figs. 7(a) and 7(b), respectively. The effect of shear deformation causes the $\sigma^2_{w,\max}$ of the Mindlin plate to be larger than that of the classical plate. However, due to the effect of rotatory inertia, $\sigma^2_{m_x,\max}$ of the Mindlin plate is smaller than that of the classical plate. Fig. 8(a) shows that $(X_\sigma)_W$ of the Mindlin plate is always near the mid-point of either the first span or the third span. However, Fig. 8(b) shows that $(X_\sigma)_{m_x}$ is near the left side of one simply supported edge.

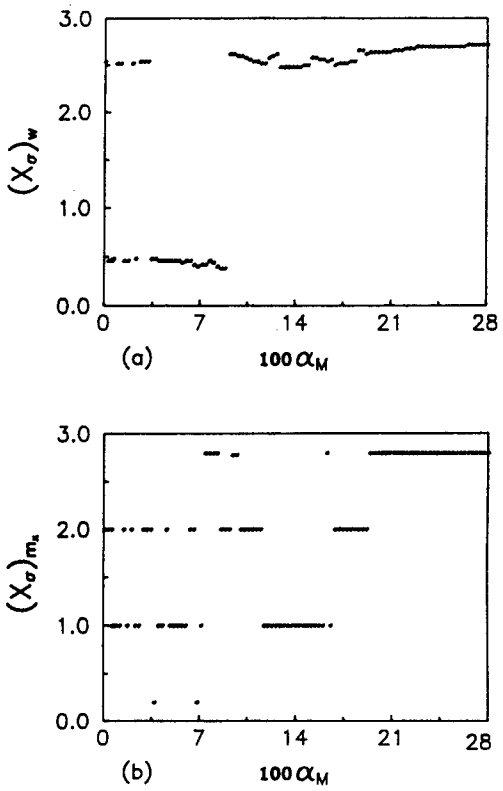


Fig. 8. (a) The $(X_\sigma)_w-\alpha_M$ distribution and (b) the $(X_\sigma)_{m_x}-\alpha_M$ distribution of a three-span Mindlin plate due to a white noise process.

3. Exponential

The effects of three ω_g (=0.01 v, 0.3 v) values of an exponential process on the $\sigma^2_{w,\max}-\alpha_M$ distribution and the $\sigma^2_{m_x,\max}-\alpha_M$ distribution of a three-span Mindlin plate are displayed in Figs. 9(a) and 9(b), respectively. Both figures show that the parameter ω_g of the process has an obvious effect on reducing both $\sigma^2_{w,\max}$ and $\sigma^2_{m_x,\max}$ of the plate, especially as the velocity ratio is near α_σ.

4. Exponential cosine

The effects of two kinds of exponential cosine processes (ω_g=0, ω_0=0.5 ω_{11}; ω_g=0.3 v, ω_0=0.5 ω_{11}) on the $\sigma^2_{w,\max}-\alpha_M$ distribution and the $\sigma^2_{m_x,\max}-\alpha_M$ distribution of a three-span Mindlin plate are displayed in Figs. 10(a) and 10(b), respectively. It can be seen that the parameter ω_g has an apparent effect on reducing both absolute $\sigma^2_{w,\max}$ and $\sigma^2_{m_x,\max}$. The α_σ presented in Fig. 10(a) is smaller than that of Fig. 9(b). This finding suggests that the α_σ is determined only by the parameter ω_0.

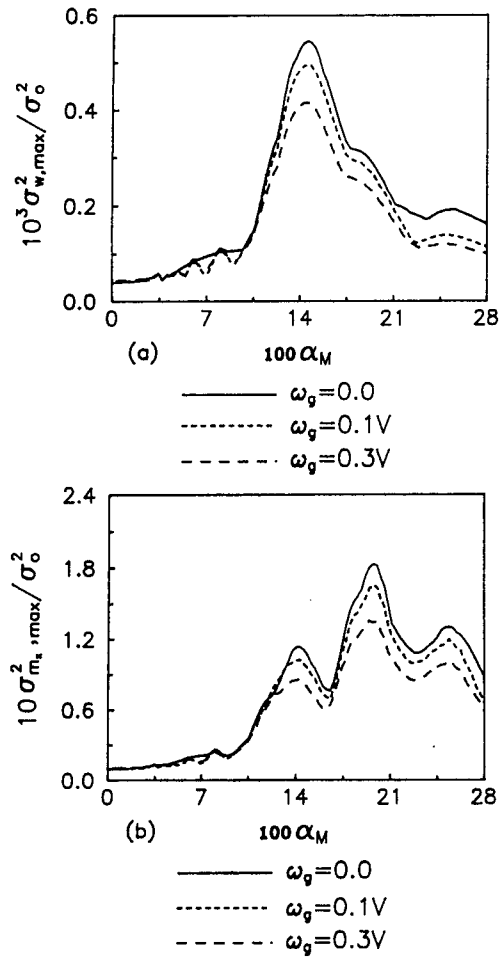


Fig. 9. Comparisons of three different ω_g effects of an exponential process on (a) the $\sigma_{w,\max}-\alpha_M$ distribution and (b) the $\sigma_{m_x,\max}-\alpha_M$ distribution of a three-span Mindlin plate.

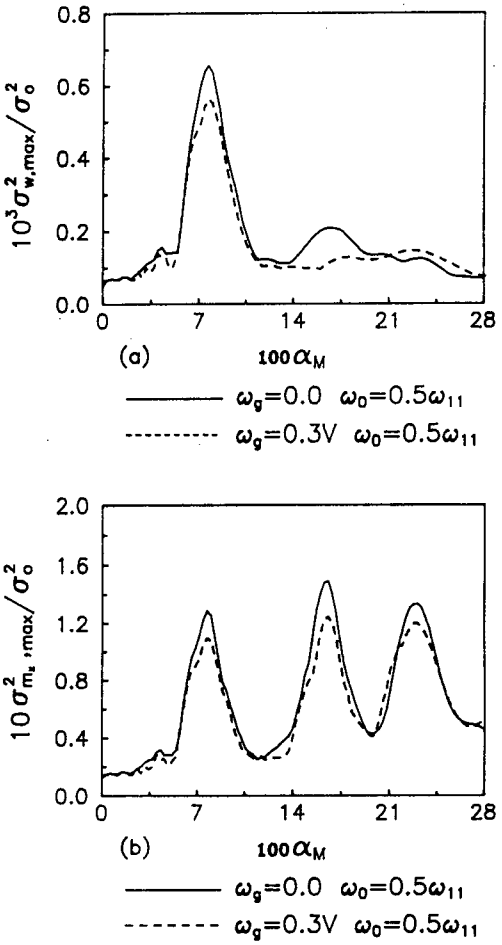


Fig. 10. Comparisons of two different ω_g effects of an exponential cosine process (ω_0=0.5 ω_{11}) on (a) the $\sigma_{w,\max}-\alpha_M$ distribution and (b) the $\sigma_{m_x,\max}-\alpha_M$ distribution of a three-span Mindlin plate.

5. Cosine

The effects of two ω_0 (= ω_{11},0.5 ω_{11}) values of the cosine process on the $\sigma^2_{w,\max}-\alpha_M$ distribution and the $\sigma^2_{m_x,\max}-\alpha_M$ distribution of a three-span Mindlin plate are displayed in Figs. 11(a) and 11(b), respectively. The plate is subjected to a quasi-steady state loading as the velocity of the load approaches zero. Both $\sigma^2_{w,\max}$ and $\sigma^2_{m_x,\max}$ will, consequently, be infinite due to the plate in resonance as $\omega_0=\omega_{11}$ and $\alpha_M=0$. However, both $\sigma^2_{w,\max}$ and $\sigma^2_{m_x,\max}$ will be finite for any α_M value expect for the case of $\omega_0=\omega_{11}$. A rapidly moving load implies a short duration of the forced vibration of the plate. The value of cosine function does not abruptly change as the loading time is short. Therefore, $\sigma^2_{w,\max}$ approaches a constant value for the load moving at a supercritical speed. Moreover, the greater ω_0 ($\leq\omega_{11}$) value of the process requires a longer duration of load moving on the plate to cause the extreme values of both $\sigma^2_{w,\max}$ and $\sigma^2_{m_x,\max}$. The above phenomena indicate that the greater ω_0 ($\leq\omega_{11}$) implies smaller α_σ.

The effects of the span number on the $\sigma^2_{w,\max}-\alpha_M$ distribution and the $\sigma^2_{m_x,\max}-\alpha_M$ distribution of a multi-span Mindlin plate due to a moving load with a variance of cosine process (ω_0=0.5 ω_{11}) are displayed in Figs. 12(a) and 12(b), respectively. The frequency (ω_0=0.5 ω_{11}) is smaller than all modal frequencies of the plate. Therefore, this cosine process can be regarded as a constant variance process. Consequently, both tendencies of the $\sigma^2_{w,\max}-\alpha_M$ (or $\sigma^2_{m_x,\max}-\alpha_M$) distribution and the $\langle w\rangle_{\max}-\alpha_M$ (or $\langle m_x\rangle_{\max}-\alpha_M$) distribution are very similar.

The effects of two different plate theories on the

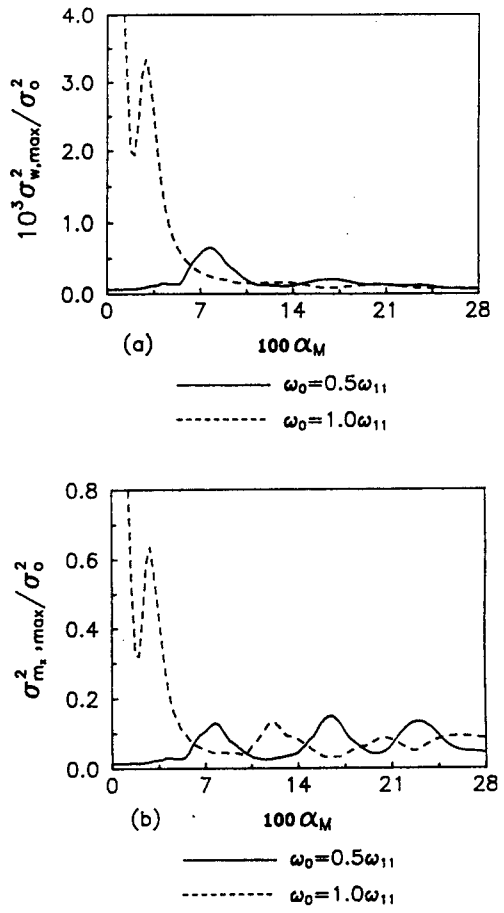


Fig. 11. Comparisons of two different ω_0 effects of a cosine process on (a) the $\sigma_{w,\max}$–α_M distribution and (b) the $\sigma_{m_x,\max}$–α_M distribution of a three-span Mindlin plate.

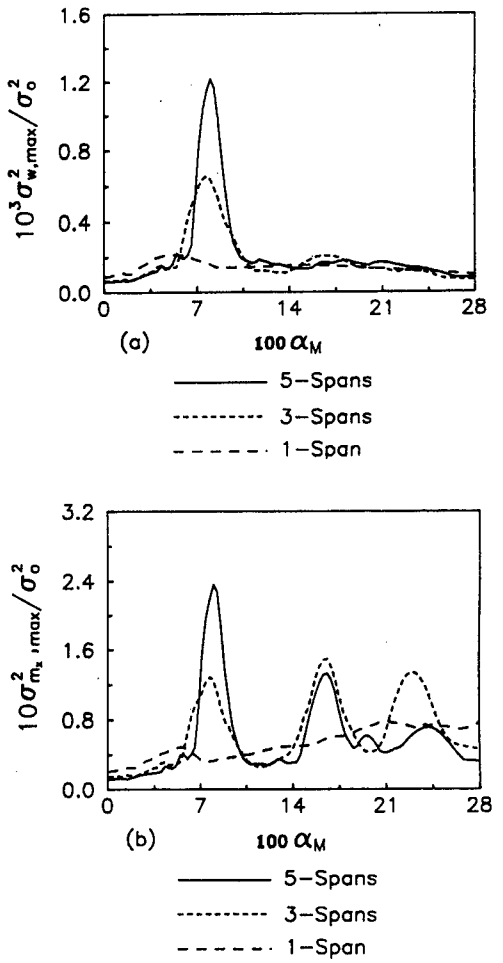


Fig. 12. Span number effect on (a) the $\sigma_{w,\max}$–α_M distribution and (b) the $\sigma_{m_x,\max}$–α_M distribution of a multi-span Mindlin plate due to a cosine process (ω_0=0.5 ω_{11}).

$\sigma^2_{w,\max}$–α_M distribution and the $\sigma^2_{m_x,\max}$–α_M distribution of a three-span plate due to a moving load with a variance of cosine process (ω_0=0.5 ω_{11}) are shown in Figs. 13(a) and 13(b), respectively. The first modal frequency of the Mindlin plate is smaller than that of the classical plate. The frequency ω_0 is closer to the first modal frequency of the Mindlin plate than that of the classical plate. Accordingly, both $\sigma^2_{w,\max}$ and $\sigma^2_{m_x,\max}$ of the Mindlin plate due to the load moving at a low speed are greater than those of the classical plate. However, the Mindlin plate has a low α_σ. The periodicity of the variance of load causes both $(X_\sigma)_W$–α_M and $(X_\sigma)_{w_x}$–α_M distributions of the plate to be different from those of a white noise process and of the mean values. The $(X_\sigma)_W$ – α_M distribution of the three-span Mindlin plate displayed in Fig. 14(a) indicates that $\sigma^2_{w,\max}$ always appears in close proximity to the mid-point of each span. However, $\sigma^2_{m_x,\max}$ may occur at the mid-point of each span or on the left side of either the second supported edge or the third supported edge, as indicated in Fig. 14(b).

V. CONCLUSIONS

The maximum mean value of transverse deflection of a multi-span Mindlin plate due to a random load moving at a constant velocity always appears in close proximity to the mid-point of either the first span or the last span. For the white noise process, both the maximum variance of transverse deflection during the motion of the load and the maximum variance of the moment of the plate during the motion of the load decrease as the velocity increases. The maximum variance of transverse deflection due to the white noise process always appears near the mid-point

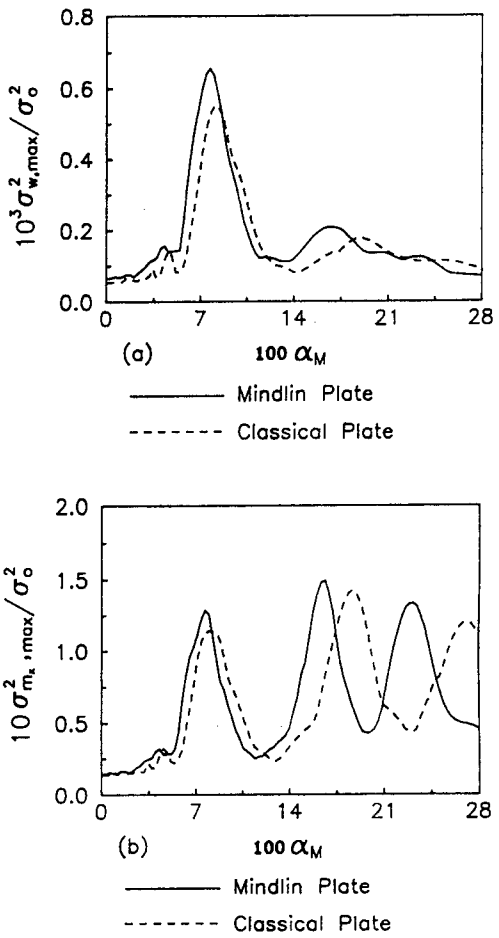


Fig. 13. Effects of two different plate theories on (a) the $\sigma_{w,max}$–α_M distribution and (b) the $\sigma_{m_x,max}$–α_M distribution of a three-span plate due to a cosine process (ω_0=0.5 ω_{11}).

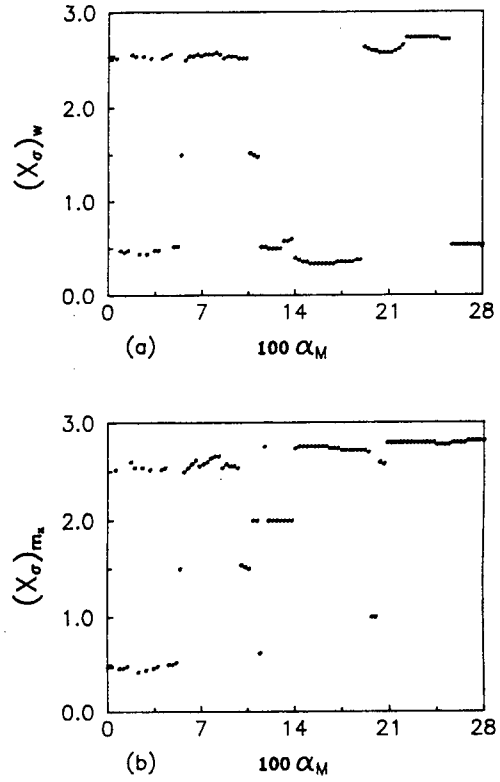


Fig. 14. (a) The $(X_\sigma)_w$–α_M distribution and (b) the $(X_\sigma)_{m_x}$–α_M distribution of a three-span Mindlin plate due to a cosine process (ω_0=0.5 ω_{11}).

of the first span or the last span of the plate. A rapidly moving load with the variance of a cosine function will not induce significant variances of deflection and moment of the plate.

ACKNOWLEDGMENTS

This study was supported by the National Science Council, R.O.C., under contract No. NSC85-2212-E006-112. This support is gratefully acknowledged.

NOMENCLATURE

a, b, h	length, width and thickness of a one-span plate
D, E, G	bending rigidity, Young's modulus and shear modulus of the plate
$\overline{m_{\bar{x}}}, \overline{m_{\bar{y}}}, \overline{m_{\bar{x}\bar{y}}}$, $\overline{q_{\bar{x}}}, \overline{q_{\bar{y}}}$	moments of the plate transverse shear forces of the plate
r	ratio of thickness to one span length
$\overline{t}$	time
$\overline{v}$	velocity
$\overline{w}$	transverse deflection of the plate
$\overline{x}, \overline{y}$	in-plane coordinates of the plate
α_M	velocity ratio
κ	shear coefficient
λ	ratio of one span length to width of the plate
μ	Poisson's ratio of the plate
$\overline{\omega_{ij}}$	the ijth modal frequency of the plate
$\overline{\psi_{\bar{x}}}, \overline{\psi_{\bar{y}}}$	rotatory angles of the cross section of the plate

REFERENCES

1. Adler, A.A. and H. Reismann, "Moving Loads on An Elastic Plate Strip," *Journal of Applied Mechanics*, Vol. 41, No. 3, pp. 713-718 (1974).

2. Cowper, G.R., "The Shear Coefficient in Timoshenko's Beam Theory," *Journal of Applied Mechanics*, Vol. 33, pp. 335-340 (1966).
3. Crandall, S.H. and A. Yildiz, "Random Vibration of Beams," *Journal of Applied Mechanics,* Vol. 29, pp. 267-275 (1962).
4. Eringen, A.C., "Response of Beams and Plates to Random Loads," *Journal of Applied Mechanics*, Vol. 24, pp. 46-52 (1957).
5. Frýba, L., "Non-stationary Response of a Beam to a Moving Random Force," *Journal of Sound and Vibration*, Vol. 46, No. 3, pp. 323-338 (1976).
6. Iwankiewicz, R. and P. Sniady, "Vibration of a Beam under a Random Stream of Moving Forces," *Journal of Structural Mechanics*, Vol. 12, pp. 13-26 (1984).
7. Knowles, J.K., "A Note on the Response of a Beam to a Random Moving Force," *Journal of Applied Mechanics*, Vol. 37, No. 4, pp. 1192-1194 (1970).
8. Love, A.E.H., *A Treatise on the Mathematical Theory of Elasticity,* Dover Publications, New York (1944).
9. Mindlin, R.D., "Influence of Rotatory Inertia and Shear on Flexural Motions of Isotropic Elastic Plates," *Journal of Applied Mechanics*, Vol. 18, pp. 31-38 (1951).
10. Raske, T.F. and A.L. Schlack, "Dynamic Response of Plates due to Moving Load," *Journal of Acoustical Society of America*, Vol. 42, No. 3, pp. 625-635 (1967).
11. Ricciardi, G., "Random Vibration of Beam under Moving Loads," *Journal of Engineering Mechanics*, Vol. 120, No. 11, pp. 2361-2380 (1994).
12. Sniady, P., "Vibration of a Beam due to a Random Stream of Moving Forces with Random Velocity," *Journal of Sound and Vibration*, Vol. 97, pp. 23-33 (1984).
13. Wang, R.T. and T.Y. Lin, "Vibration of Multispan Mindlin Plates to a Moving Load," *Journal of the Chinese Institute of Engineers*," Vol. 19, No. 4, pp. 121-123 (1996).

Discussions of this paper may appear in the discussion section of a future issue. All discussions should be submitted to the Editor-in-Chief.

Manuscript Received: May 26, 1997
Revision Received: Jan. 24,1998
and Accepted: Feb. 9, 1998

多跨距 Mindlin 板結構承受移動負載之隨機振動分析

王榮泰　林長源

國立成功大學工程科學系

摘　要

本文主要是以模態法分析多跨距Mindlin板結構承受等速移動負載之隨機動態響應。考慮負載爲靜定隨機過程，並以四種變異相關函數：白色干擾、指數遞減、指數遞減之餘弦與餘弦等，進行研究。探討負載之速度和隨機變異函數與板之跨距數對於結構內位移與彎矩響應之最大平均值與最大變異數之影響，並和古典板之結果作比較。

關鍵詞：多跨距、Mindlin 板、移動負載、平均值與變異數。

EFFECT OF S/A RATIO ON THE ELASTIC MODULUS OF CEMENT-BASED MATERIALS

Chung-Chia Yang*
Institute of Materials Engineering
National Taiwan Ocean University
Keelung, Taiwan 202, R.O.C.

Ran Huang
Department of Harbor and River Engineering
National Taiwan Ocean University
Keelung, Taiwan 202, R. O. C.

Key Words: elastic modulus, aggregate, concrete, S/A ratio.

ABSTRACT

In order to investigate the elastic modulus of aggregate and the effect of fine aggregate content on the elastic modulus of concrete, cylindrical specimens (ϕ100×200 mm) with different volume ratios (*S/A*, fine aggregate volume/ total aggregate volume) and various water/ cement ratios were cast and tested. Both single-inclusion and double-inclusion models were applied to predict the elastic moduli of two-phase and three-phase cement-based composite materials, respectively. The elastic moduli of sand and coarse aggregate were derived from the experimental results using the theoretical models. In addition, the elastic modulus of concrete is not significantly influenced by the *S/A* ratio for a constant aggregate volume.

I. INTRODUCTION

A composite material can be defined as a combination of at least two different materials. Usually the properties of a multiphase composite are different from the properties of the original components. It is appropriate to consider concrete as a cement-based composite which consists of aggregate embedded in a matrix of hydrated cement paste.

Concrete researchers have investigated the relationship between the elastic modulus and aggregate volume fraction. Stock *et al.* [15] investigated the effect of aggregate volume upon elastic modulus of concrete and theoretical predictions were compared with experimental data based upon the mixture laws. Zhou *et. al* [19] applied the composite model to study the effect of coarse aggregate properties upon the elastic modulus of high performance concrete. By considering concrete as a two-phase material, Aitcin and Mehta [1], and Baalbaki *et al.* [3] demonstrated that the elastic modulus of concrete is influenced by the elastic properties and volume fraction of aggregates. Hirsch [9] derived an equation to express the elastic modulus of concrete by employing an empirical constant, and also reported the experimental results of elastic moduli of concretes with different aggregates. For high performance concrete, segregation may be reduced by increasing the fine aggregate content and adding superplasticizer [10].

The overall mechanical behavior of composite materials has been extensively studied. The elastic

*Correspondence addressee

moduli of the concrete composite materials are controlled by the properties and volume fraction of the matrix and inclusion. In previous studies, Voigt's [16] approximation yielded the upper bound and the Reuss's [14] approximation obtained the lower bound of the average elastic moduli. Hashin and Shtrikman [8] proposed the variational principle to find bounds of the elastic moduli of composite materials which appeared better than the Voigt and Reuss bounds. Hansen [7] developed a mathematical model to predict the elastic moduli of composite materials based on the elastic modulus and volume fraction of the component. Mori and Tanaka [11] applied the concept of average field to analyze macroscopic properties of composite materials. The average field in a body contains inclusions with eigenstrain. Later, the shape effect of dispersoids was introduced in Eshelby's [5] method to assess the properties of composite materials. Recent developments of evaluating overall elastic modulus and overall elastic-plastic behavior was reviewed by Mura [12], Nemat-Nasser and Hori [13]. Yang and Huang proposed a single-inclusion model [17] as well as a double-inclusion model [18] for approximating elastic modulus of concrete by employing both Mori-Tanaka Theory and Eshelby's Method.

The published literature contains little information on the influence of fine aggregate content on the elastic modulus of concrete. In this study, the elastic moduli of cement paste, mortar and concrete (concrete without fine aggregate was included) were obtained in the laboratory. By considering cement paste as the matrix, the single-inclusion model [17] was used to evaluate the equivalent elastic moduli of fine aggregate and coarse aggregate. In addition, the double-inclusion model [18] was used to predict the elastic modulus of concrete by considering it as a three phase composite.

II. EXPERIMENTAL PROGRAM

In this study, single-inclusion cement-based material was considered a two-phase composite material in which sand particles or coarse aggregate were embedded in the matrix. Concrete was a three-phase composite material, i.e. besides the matrix (cement paste), fine aggregate and coarse aggregate were the two inclusions.

1. Cement paste (matrix)

Cement paste specimens were made of type I cement, silica fume, type F superplasticizer, and water. Three different water/cement ratios (w/c=0.26, 0.30, and 0.34) were selected and the mix design is given in Table 1. Cylindrical specimens (ϕ100×200 mm) were cast and cured in water until the time of testing. At the age of 28 days, the elastic moduli of the specimens were measured according to the specifications of ASTM C-469-81. All cylinders were ground and polished before testing to achieve a smooth end surface. A testing machine of 100 kN-load capacity was used.

Table 1. Mix Proportions of Cement Paste (kg/m^3)

Materials	w/c=0.26	w/c=0.3	w/c=0.34
water	392	436	473
Cement	1527	1428	1341
Silica fume (SF)	153	143	134
Superplasticizer (SP)	44	36	28

2. Single-inclusion cement-based materials (two-phase composite)

In this study, two types of single-inclusion cement-based materials were used. One was cement paste with sand and the other was cement paste with crushed stone. The mix design is given in Table 2. Notation for the specimens is such that the first letter indicates two different aggregates S and R, and the second letter A, B, or C indicates three different water/cement ratios 0.26, 0.30, and 0.34, respectively. The aggregate volume ratio (aggregate volume/concrete volume, *A/C*) of 58% was selected. The cylindrical specimens (ϕ100×200 mm) were cast and cured for each batch of single-inclusion cement-based materials. At the age of 28 days, the elastic moduli of the specimens were measured according to the specification of ASTM C-469-81.

3. Concrete (three-phase composite)

Concrete was made of Type I cement, silica fume, superplasticizer, water, natural sand and crushed stone. Three water/cement ratios (w/c=0.26, 0.30, and 0.34) and five different volume ratios of fine aggregate (volume ratio of fine aggregate/ total aggregate, *S/A*=0.3, 0.4, 0.5, 0.6, and 0.7) were considered in the mix proportions. The concrete mix design is given in Table 3. Notation for the specimens is such that the first letter A, B, or C indicates three different water/cement ratios 0.26, 0.30, and 0.34, respectively. The second number is the volume ratio of the coarse aggregate. Mortar and concrete cylinders (ϕ100×200 mm) were cast and cured. At the age of 28 days, the elastic moduli and compressive strength of the specimens were measured according to ASTM C 469-81 and ASTM C 39-81, respectively.

Table 2. Mix Proportions of Single Inclusion Cement Based Materials (kg/m³)

Mix No.	Water	Cement	SF	SP	Fine Aggregate	Coarse Aggregate
SA	161.3	627.8	62.8	18.2	1516.1	0
SB	179.1	587.1	58.7	14.7	1516.1	0
SC	194.7	551.4	55.1	11.6	1516.1	0
RA	161.3	627.8	62.8	18.2	0	1537.0
RB	179.1	587.1	58.7	14.7	0	1537.0
RC	194.7	551.4	55.1	11.6	0	1537.0

Table 3. Mix Proportions of Concrete (kg/m³)

Mix No.	Water	Cement	SF	SP	Coarse Aggregate	Fine Aggregate
A3					1075.9	454.8
A4	161.3	627.8	62.8	18.2	922.2	606.4
A5					768.5	758.1
A6					614.8	909.7
A7					461.1	1061.3
B3					1075.9	454.8
B4	179.1	587.1	58.7	14.7	922.2	606.4
B5					768.5	758.1
B6					614.8	909.7
B7					461.1	1061.3
C3					1075.9	454.8
C4	194.7	551.4	55.1	11.6	922.2	606.4
C5					768.5	758.1
C6					614.8	909.7
C7					461.1	1061.3

III. THEORETICAL BACKGROUND

In this study, mortar and single-inclusion cement-based materials were considered a two-phase (cement paste and fine aggregate or coarse aggregate) composite material and concrete was considered a three-phase (cement paste, fine aggregate, and coarse aggregate) composite material. The inclusions were randomly embedded in an infinite matrix. Calculations were divided into two stages. In the first stage, the equivalent elastic modulus of the fine aggregate was calculated by the single-inclusion model for a two-phase composite. Secondly, the double-inclusion model for a three-phase composite was used to calculate the elastic modulus of coarse aggregate.

1. Single-inclusion model

The theoretical model is based on Mori-Tanaka Theory and Eshelby's Method in which the stress disturbance in the applied compressive stress, due to inhomogeneities, can be simulated by the eigenstress caused by the fictitious misfit strain. The fictitious misfit strain (eigenstrain), was introduced to simulate the inhomogeneity effect. This model can provide an evaluation of average elastic relationships of cement-based materials with spherical inhomogeneities. The overall average elastic moduli of cement-based material $\bar{\underline{C}}$ were given by [17]

$$\bar{\underline{C}} = \{\underline{C}^{-1} + f[\{1-f\}(\underline{C}^* - \underline{C})\underline{S} - f(\underline{C} - \underline{C}^*) + \underline{C}\}^{-1}]^{-1} \cdot (\underline{C} - \underline{C}^*)\underline{C}^{-1}\}, \quad (1)$$

where $\underline{C}$ and $\underline{C}^*$ are the elastic moduli tensor of matrix and aggregate, respectively. f is the volume ratio of inclusion, and $\underline{S}$ is the Eshelby tensor. The Eshelby tensor is a function of the geometry of the inclusion and Poisson's ratio of the matrix (see Appendix A).

2. Double-inclusion method

The double-inclusion method was applied to

calculate the equivalent elastic modulus of coarse aggregate. The overall elastic moduli of concrete composite materials were investigated in this study by employing the theory of micromechanics. The inclusions were divided into two groups: fine aggregate and coarse aggregate. The overall elastic moduli of the concrete composite materials were given as a function of properties and volume ratio of the following three components: fine aggregate, coarse aggregate, and cement paste. In previous work [18], a composite material was simulated by a homogeneous material with uniform stiffness $\underset{\sim}{C}$ and distributing eigenstrains $\underset{\sim}{\varepsilon}_1^*$ in the domain of fine aggregate and $\underset{\sim}{\varepsilon}_2^*$ in the domain of coarse aggregate, respectively. The distributing eigenstrains $\underset{\sim}{\varepsilon}_1^*$ and $\underset{\sim}{\varepsilon}_2^*$ were calculated as

$$\left\langle \underset{\sim}{\varepsilon}_1^* \right\rangle = \alpha \underset{\sim}{\sigma}^0, \tag{2}$$

$$\left\langle \underset{\sim}{\varepsilon}_2^* \right\rangle = \beta \underset{\sim}{\sigma}^0, \tag{3}$$

α and β are shown in the Appendix B. $\underset{\sim}{\sigma}^0$ is an applied uniform stress. The average elastic moduli tensor of concrete composite materials, $\bar{\underset{\sim}{C}}$, for a three-phase composite material is given by

$$\bar{\underset{\sim}{C}} = (\underset{\sim}{C}^{-1} + f_1\alpha + f_2\beta)^{-1}, \tag{4}$$

where f_1 and f_2 are the volume ratio of fine aggregate and coarse aggregate, respectively.

IV. RESULTS AND DISCUSSIONS

The characteristics of natural mineral aggregate are derived from mineralogical composition of the bedrock. The mineralogical composition of aggregate affects its elastic modulus, which in turn influences the elastic modulus of hardened concrete. Many researchers [9, 2, 3, 4, 6, 19] have investigated the elastic modulus of aggregate. Even for the same mineralogical composition, the aggregate elastic modulus may be different.

Single-inclusion cement-based materials can be considered a two-phase composite material, i.e. besides the matrix, fine aggregate or coarse aggregate is the inclusion. Figs. 1 and 2 illustrate the elastic moduli vs. water/cement ratio curves for the single-inclusion cement-based materials with a constant aggregate volume ratio of 0.58. Test results show that the elastic modulus of single-inclusion cement-based materials and cement paste decreases as water-cement ratio increases.

Poisson's ratio of cement paste and single-inclusion cement-based materials was assumed to be 0.2 (by changing the Poisson's ratio from 0.16 to 0.3,

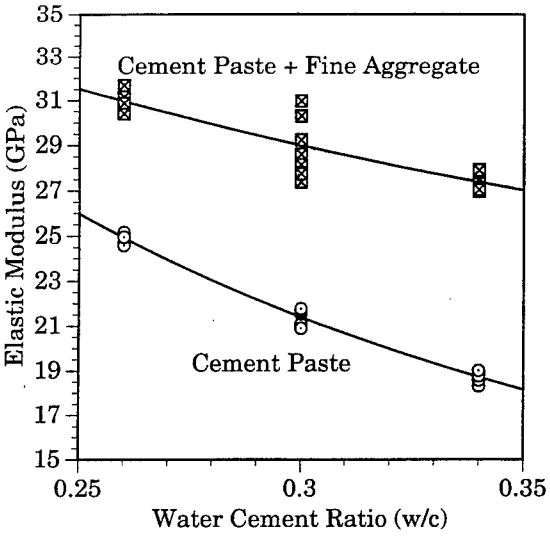


Fig. 1. Elastic modulus of mortar vs. w/c curve.

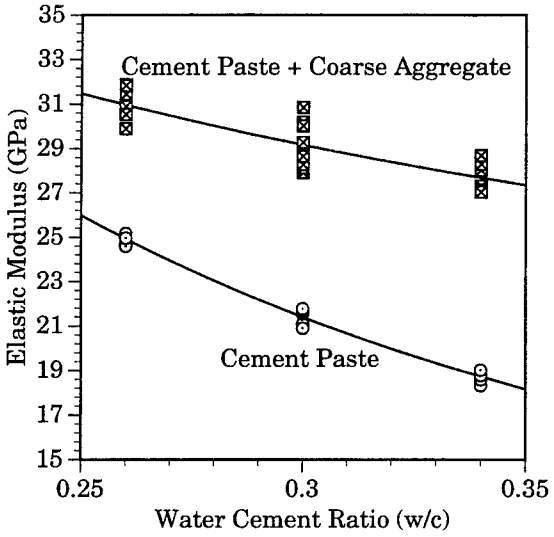


Fig. 2. Elastic modulus of single inclusion cement based materials vs. w/c curve.

the computed elastic moduli were not significantly affected as tabulated in Table 4) for the computation of the elastic modulus tensors of the matrix and the composite. The volume ratio of sand or crushed stone is 0.58. The elastic moduli of cement paste and single-inclusion cement-based materials were experimentally determined and are presented in Table 5. Eq. (1) was used to calculate the elastic modulus of the fine aggregate and coarse aggregate based on the single-inclusion model and experimental results. The computed elastic moduli of the fine aggregate and

Table 4. The computed elastic moduli (Poisson's ratio from 0.16 to 0.3)

Poisson's ratio of Paste	Poisson's ratio of Sand	Poisson's ratio of Gravel	Elastic Modulus of Concrete (GPa)
0.16	0.2	0.2	29.219
0.18	0.2	0.2	29.202
0.20	0.2	0.2	29.195
0.22	0.2	0.2	29.199
0.24	0.2	0.2	29.213
0.26	0.2	0.2	29.238
0.28	0.2	0.2	29.275
0.30	0.2	0.2	29.325
0.2	0.16	0.2	29.204
0.2	0.18	0.2	29.197
0.2	0.20	0.2	29.195
0.2	0.22	0.2	29.197
0.2	0.24	0.2	29.204
0.2	0.26	0.2	29.216
0.2	0.28	0.2	29.233
0.2	0.30	0.2	29.255
0.2	0.2	0.16	29.204
0.2	0.2	0.18	29.197
0.2	0.2	0.20	29.195
0.2	0.2	0.22	29.197
0.2	0.2	0.24	29.204
0.2	0.2	0.26	29.216
0.2	0.2	0.28	29.233
0.2	0.2	0.30	29.255

Young's modulus of cement paste: E_m=21.43 *GPa*
Young's modulus of fine aggregate: E_s=36.73 *GPa*
Young's modulus of coarse aggregate: E_g=36.95 *GPa*
Volume fraction of cement paste: 0.42
Volume fraction of sand: 0.29
Volume fraction of gravel: 0.29

Table 5. Elastic moduli of cement paste, mortar, and fine aggregate.

Designation	Elastic Modulus, GPa		
	*Cement Paste (matrix) (Experimental)	**Cement Based Materials (two-phase composite) (Experimental)	Aggregate (inclusion) (Theoretical)
SA	24.91	31.09	36.63
SB	21.43	29.08	36.58
SC	18.71	27.50	36.97
			Average=36.73
RA	24.91	31.037	36.51
RB	21.43	29.167	36.78
RC	18.71	27.730	37.55
			Average=36.95

*Average of seven specimens
**Average of eight specimens

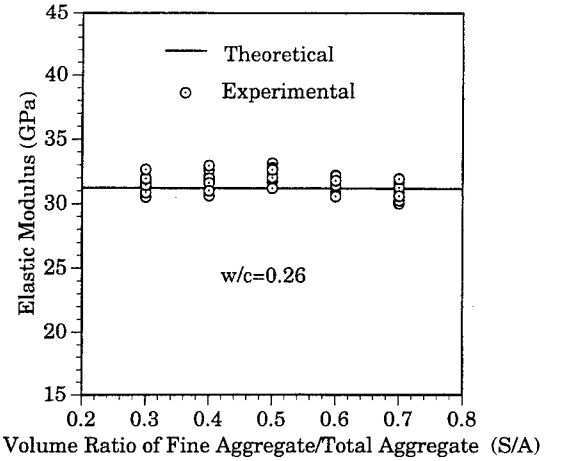


Fig. 3 Elastic modulus of concrete vs. volume ratio of fine aggregate/total aggregate.

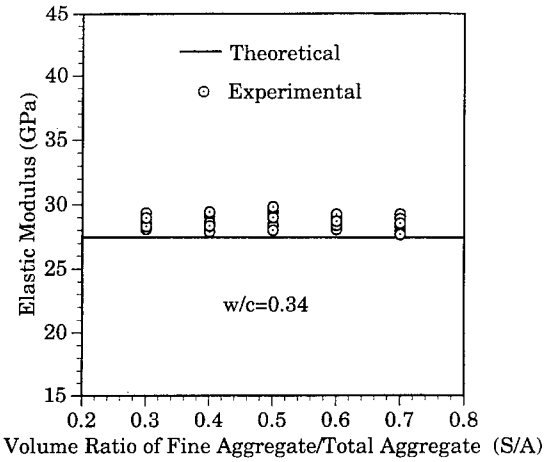


Fig. 5. Elastic modulus of concrete vs. volume ratio of fine aggregate/total aggregate.

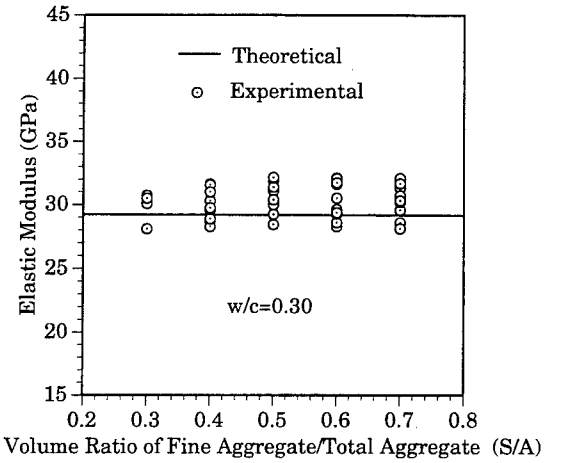


Fig. 4. Elastic modulus of concrete vs. volume ratio of fine aggregate/total aggregate.

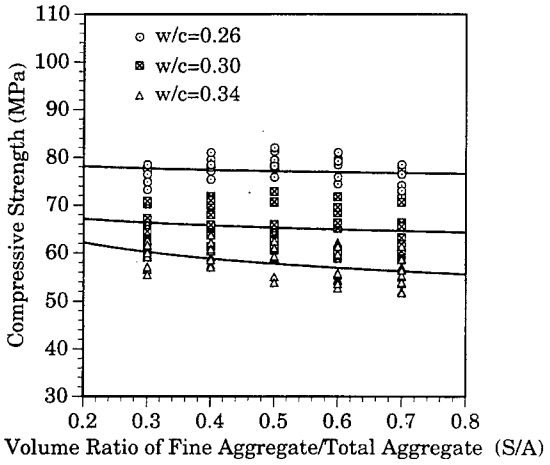


Fig. 6. Compressive strength of concrete vs. volume ratio of fine aggregate/total aggregate.

coarse aggregate are tabulated in Table 5. The average computed elastic modulus is 36.73 GPa for fine aggregate and 36.95 GPa for coarse aggregate, respectively.

A three-phase composite with spherical inclusions was also taken into account. In the theoretical approach, the elastic moduli of cement paste and aggregate, listed in Table 5, were used. Poisson's ratio of cement paste, fine aggregate, and coarse aggregate was assumed to be 0.2. The volume ratios of fine aggregate and coarse aggregate were calculated from Table 3. The elastic modulus of concrete was calculated from Eq (4) based on the elastic moduli, Poisson's ratios, and volume ratios of cement paste, fine aggregate, and coarse aggregate. Figs. 3, 4 and 5 show the relationships between the elastic modulus of concrete and the volume ratio of fine aggregate (S/A) for water/cement ratios of 0.26, 0.30, and 0.36, respectively. The corresponding theoretical results are also illustrated in the Figures. Since the elastic modulus of fine aggregate and coarse aggregate is very close (see Table 5) and the total volume ratio of the aggregate is constant (A/C), the calculated concrete elastic moduli are almost the same (see Figs. 3, 4, and 5). In Figs. 3, 4, and 5, as the total volume of aggregate is constant, the experimental concrete elastic moduli are not significantly influenced by the S/A ratios.

Figure 6 displays the relationship between the compressive strength of concrete and the S/A

ratio for specimens with various water/cement ratios. It appears that compressive strength of concrete is not significantly affected by the *S/A* ratio in this study.

V. CONCLUSIONS

The elastic moduli of single-inclusion cement-based materials are influenced by the elastic properties and the volume ratio of the matrix (cement paste); fine aggregate or coarse aggregate. Based on the single-inclusion model, the average elastic modulus of fine aggregate and coarse aggregate can be computed from the composite properties. In this study, the average estimated elastic moduli of fine aggregate and coarse aggregate are 36.73 GPa and 36.95 GPa, respectively. Since the elastic moduli of fine aggregate and coarse aggregate are very close, the concrete elastic moduli are not significantly affected by the *S/A* ratio.

ACKNOWLEDGMENT

The financial support of National Science Council under the grants NSC 86-2211-E-019-002 is gratefully appreciated.

NOMENCLATURE

A	volume of total aggregate, m^3
c	weight of cement, kg
C	volume of concrete, m^3
$\underset{\sim}{C}$	elastic moduli of matrix, Pa
$\underset{\sim}{C}^*$	elastic moduli of inclusion, Pa
$\underset{\sim}{C}_1^*$	elastic moduli of fine aggregate, Pa
$\underset{\sim}{C}_2^*$	elastic moduli of coarse aggregate, Pa
$\underset{\sim}{\bar{C}}$	elastic moduli of composite, Pa
f	volume ratio of inclusion
f_1	volume ratio of fine aggregate
f_2	volume ratio of coarse aggregate
S	volume of sand, m^3
$\underset{\sim}{I}$	unit tensor
$\underset{\sim}{S}$	Eshelby tensor
w	weight of water, kg
v	Poisson's ratio
$\underset{\sim}{\sigma}^0$	applied uniform stress, Pa
$\underset{\sim}{\varepsilon}_1^*$	eigenstrain of fine aggregate
$\underset{\sim}{\varepsilon}_2^*$	eigenstrain of coarse aggregate

REFERENCES

1. Aïtcin, P.C., and P.K. Mehta, "Effect of Coarse-aggregate Characteristics on Mechanical Properties of High-strength Concrete," *ACI Materials Journal*, Vol. 87, No. 2, pp. 103-107 (1990).
2. Anson, M. and K. Newman, "The Effect of Mix Proportions and Method of Testing on Poisson's Ratio for Mortars and Concretes," *Magazine of Concrete Research*, Vol. 18, No. 56, pp. 115-130 (1966).
3. Baalbaki, W., B. Benmokrane, O. Chaallal, and P.C. Aïtcin, "Influence of Coarse Aggregate on Elastic Properties of High-performance Concrete," *ACI Materials Journal*, Vol. 88, No. 5, pp. 499-503 (1991).
4. Baalbaki, P.C. Aïtcin, and G. Ballivy, "On Predicting Modulus of Elasticity in High-Strength Concrete," *ACI Materials Journal*, Vol. 89, No. 5, pp. 517-520 (1992).
5. Eshelby, J.D., "The Determination of the Elastic Field of an Ellipsoidal Inclusion, and Related Problems" *Proceeding Royal Society*, A241, pp. 376-396 (1957).
6. Giaccio, G., C. Pocco, D. Violini, J. Zappitelli, and R. Zerbino, "High-Strength Concrete Incorporating Different Coarse Aggregates," *ACI Materials Journal* , Vol. 89, No. 3, pp. 242-246 (1992).
7. Hansen, T.C., "Strength Elasticity, and Creep as Related to the Internal Structure of Concrete," Chemistry of Cement, Proceedings of the Fourth International Symposium, Monograph 43, Vol. 2, pp. 709-723 (1960).
8. Hashin, Z. and S. Shtrikman, "On Some Variational Principles in Anisotropic and Nonhomogeneous Elasticity," *Journal of the Mechanics and Physics of Solids*, Vol. 10, pp. 335-343, (1962).
9. Hirsch, T.J., "Modulus of Elasticity of Concrete Affected by Elastic Moduli of Cement Paste Matrix and Aggregate," *ACI Journal*, pp. 427-451, (1962).
10. Mehta, P.K. and P.J.M. Monteiro, *Concrete: Structure, Properties, and Materials*, Second Edition, Prentice Hall, New Jersey, N. J., pp. 256-257 (1993).
11. Mori, T. and K. Tanaka, "Average Stress in Matrix and Average Energy of Materials with Misfitting Inclusions," *Acta Metallurgica*, Vol. 21, pp. 571-574 (1973).
12. Mura, T., *Micromechanics of Defects in Solids*, Second Revised Edition, Martinus Nijhoff Publishers, The Hague, The Netherlands, pp. 364-380(1987).
13. Nemat-Nasser, S. and M. Hori, *Micromechanics: Overall Properties of Heterogeneous Materials*, Elsevier Science, Amsterdam, The Netherlands, pp. 229-246(1993).
14. Reuss, A., "Berechnung der Fliessgrenze von Mischkristallen auf Grund der Plastizität Sbedingung fÜr Einkristalle," *Zeitschrift Fur Angefwandte Mathematik Und Mechan*, Vol. 9,

pp. 49-58, (1929).
15. Stock, A.F., D.J. Hannant, and R.I.T. Wiliams, "The Effect of Aggregate Concentration upon the Strength and Modulus of Elasticity of Concrete," *Magazine of Concrete Research*, Vol. 31, No. 109, pp. 225-234 (1979).
16. Voigt, W., "Über die B eziehung zwischen den beiden Elastizität Skonst Anten Isotroper KÖrper," *Wiederbelebung Annales*, Vol. 38 pp. 573-587, (1889).
17. Yang, C.C., R. Huang, W. Yeih, and J.J. Chang, "Theoretical Approximate Elastic Moduli of Concrete Material," *The Chinese Journal of Mechanics*, Vol. 11, pp. 47-53 (1995).
18. Yang, C.C. and R. Huang, "Double Inclusion Model for Approximate Elastic Moduli of Concrete Material," *Cement and Concrete Research*, Vol. 26, No. 1, pp. 83-91 (1996).
19. Zhou, F.P., F.D. Lydon, and B.I.G. Barr, "Effect of Coarse Aggregate on Elastic Modulus and Compressive Strength of High Performance Concrete," *Cement and Concrete Research*, Vol. 25, No. 1, pp. 177-186 (1995).

APPENDIX A

Eshelby's tensor $\underset{\sim}{S}$ for sphere inclusion is listed below [12].

$$S_{1111}=S_{2222}=S_{3333}=\frac{7-5\nu}{15(1-\nu)}.$$

$$S_{1122}=S_{2233}=S_{3311}=S_{1133}=S_{2211}=S_{3322}=\frac{5\nu-1}{15(1-\nu)}.$$

$$S_{1212}=S_{2323}=S_{3131}=\frac{4-5\nu}{15(1-\nu)}.$$

APPENDIX B

The calculation of parameters α and β

$$A=[(1-f_1)\underset{\sim}{C}+f_1\underset{\sim}{C_1}^*](\underset{\sim}{S}-\underset{\sim}{I})-\underset{\sim}{C_1}^*\underset{\sim}{S}$$

$$B=[(1-f_2)\underset{\sim}{C}+f_2\underset{\sim}{C_2}^*](\underset{\sim}{S}-\underset{\sim}{I})-\underset{\sim}{C_2}^*\underset{\sim}{S}$$

$$M=(\underset{\sim}{C_1}^*-\underset{\sim}{C})(\underset{\sim}{S}-\underset{\sim}{I})$$

$$N=(\underset{\sim}{C_2}^*-\underset{\sim}{C})(\underset{\sim}{S}-\underset{\sim}{I})$$

$$\langle\underset{\sim}{\varepsilon_1}^*\rangle=-(\underset{\sim}{I}-f_1f_2A^{-1}MB^{-1}N)^{-1}A^{-1}[f_2MB^{-1}(\underset{\sim}{C_2}^*\underset{\sim}{C}-\underset{\sim}{I})+(\underset{\sim}{C_1}^*\underset{\sim}{C}-\underset{\sim}{I})]\underset{\sim}{\varepsilon}^0=\alpha\underset{\sim}{\varepsilon}^0$$

$$\langle\underset{\sim}{\varepsilon_2}^*\rangle=B^{-1}\{f_1N(\underset{\sim}{I}-f_1f_2A^{-1}MB^{-1}N)^{-1}A^{-1}\cdot[f_2MB^{-1}(\underset{\sim}{C_2}^*\underset{\sim}{C}-\underset{\sim}{I})-(\underset{\sim}{C_1}^*\underset{\sim}{C}-\underset{\sim}{I})]+(\underset{\sim}{C_2}^*\underset{\sim}{C}-\underset{\sim}{I})\}\underset{\sim}{\varepsilon}^0=\beta\underset{\sim}{\varepsilon}^0.$$

Discussions of this paper may appear in the discussion section of a future issue. All discussions should be submitted to the Editor-in-Chief.

Manuscript Received: June 17, 1997
Revision Received: Feb. 06, 1998
and Accepted: Feb. 19, 1998

S/A 比對水泥質材料彈性模數及強度之影響

楊仲家 *

國立台灣海洋大學材料工程研究所

黃 然

國立台灣海洋大學河海工程系所

摘 要

本研究中製作不同的細骨材體積比（S/A，細骨材體積／總骨材體積）及不同水灰比之圓柱試體（ϕ 100 x 200 mm）進行實驗，以探討骨材彈性模數及細骨材含量對混凝土彈性模數之影響。文中利用單置入物模式及雙置入物模式推估混凝土彈性模數。經由試驗數據與理論式節配合，本研究中可求得粗骨材及細骨材的彈性模數。研究結果顯示，當總骨材體積爲定值時，細骨材體積比(S/A)對於混凝土彈性模數的影響不顯著。

關鍵詞：彈性模數，骨材，混凝土，S/A 比。

AN EXPLANATION OF DISTANCE-DEPENDENT DISPERSION OF MASS TRANSPORT IN FRACTURED ROCK

Bih-Shan Lin, and Cheng-Haw Lee*
Department of Mineral and Petroleum Engineering,
National Cheng Kung University,
Tainan, Taiwan, 701, R.O.C.

Key Words: percolation theory, discrete fracture, dispersion.

ABSTRACT

This paper presents a stochastic, discrete fracture model to investigate the distance-dependent dispersion phenomenon in fractured rock. Under imposed boundary conditions, the dispersion behavior of particles is observed. Simulated results demonstrate that the coefficient of anisotropic dispersion tensor increases with an increasing traveled distance. The anisotropic ratio of dispersion tensor is small and almost increased linearly with migration distance at early migration. The anisotropic ratio has the trend to maintain constant or small variation after a specified migration distance. It was also noted that the plume of particles is elliptic.

I. INTRODUCTION

In recent years, investigations on solute transport in fractured rock formations have become an increasingly important topic, focusing primarily on possible subsurface contamination by leakage from radioactive and hazardous waste repositories. Previous studies on the subject of particle transport (Berkowitz and Breaster, 1991, Lee *et al.*, 1994; Lin et al., 1997) have numerically certified that the average absolute travel distance of particles $<r^2>$ is proportional to t^d, where t is the travel time and d is an exponent value (= the slope of log $<r^2>$ vs. log t). These studies indicated that the critical exponent value at a percolation threshold approximates to the theoretic value of 1.27 provided by Sahami and Imdakm (1988). However, Lin *et al.* (1997) pointed out that below the threshold, the percolation of particle transport could happen at certain conditions, dependent on the fracture geometric parameters and flow pattern. Meanwhile, in the diagram of the exponent value versus fracture parameter, a V-shape, distributed to the threshold, was found. This indicated that the dispersion behavior of particle transport in the fracture network may be related to not only the fracture geometric parameters but also the migration distance. In this paper, the new index of particle travel distance is defined to describe the distance-dependent dispersion. The dispersion phenomenon is described by using the exponent value and the index.

II. DISPERSION CALCULATION

In this paper, the displacement-moment approach is developed to estimate the relation between the dispersion coefficient and the travel distance. At first, we define the time t which is required for a particle to reach a distance r. The squared distance r^2 was calculated for a particle at various times:

$$r^2=(x-x_o)^2+(y-y_0)^2 \quad (1)$$

where x and y are the coordinates of the particle

*Correspondence addressee

at time t, and x_o and y_0 are the coordinates of the injection point, respectively.

On the other hand, for examining the anisotropy of dispersion, Schwartz and Smith (1988) and Way and McKee (1981) proposed a generalized theory of anisotropic dispersion in two-dimensional fracture networks. A straightforward procedure was provided to calculate components of the anisotropic dispersion tensor D_{ij} for the fracture network subject to flow with a specific gradient, as shown in the following.

$$D_{ij} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix}. \quad (2)$$

$$D_{xx}=(<xx>-2<xt><x>/<t>+<tt><x>^2/<t>^2)/(2<t>) \quad (3)$$

$$D_{xy}=D_{yx}=(<xy>-<xt><y>/<t>-<yt><x>/<t> +<tt><x><y>/<t>^2)/(2<t>) \quad (4)$$

$$D_{yy}=(<yy>-2<yt><y>/<t>+<tt><y>^2/<t>^2)/(2<t>). \quad (5)$$

Once D_{xx}, D_{xy} and D_{yy} have been calculated, the major and minor dispersion tensor can be expressed as (Way and Makee, 1981)

$$D_{11} = (D_{xx} + D_{yy})/2 + [(D_{xx} - D_{yy})^2 + 4D_{xy}^2]^{0.5}/2 \quad (6)$$

$$D_{22} = (D_{xx} + D_{yy})/2 - [(D_{xx} - D_{yy})^2 + 4D_{xy}^2]^{0.5}/2 \quad (7)$$

Thus , the 2-D dispersion tensors and anisotropic ratio, defined as D_{11}/D_{22}, respectively can be calculated from the stochastic description of particle spreading, and the behavior of particle transport can be observed. The anisotropic ratio D_{11}/D_{22} provides a thorough understanding of the dispersion variability of anisotropy in fracture networks when the slope of ln $<r^2>$ vs. ln t was known.

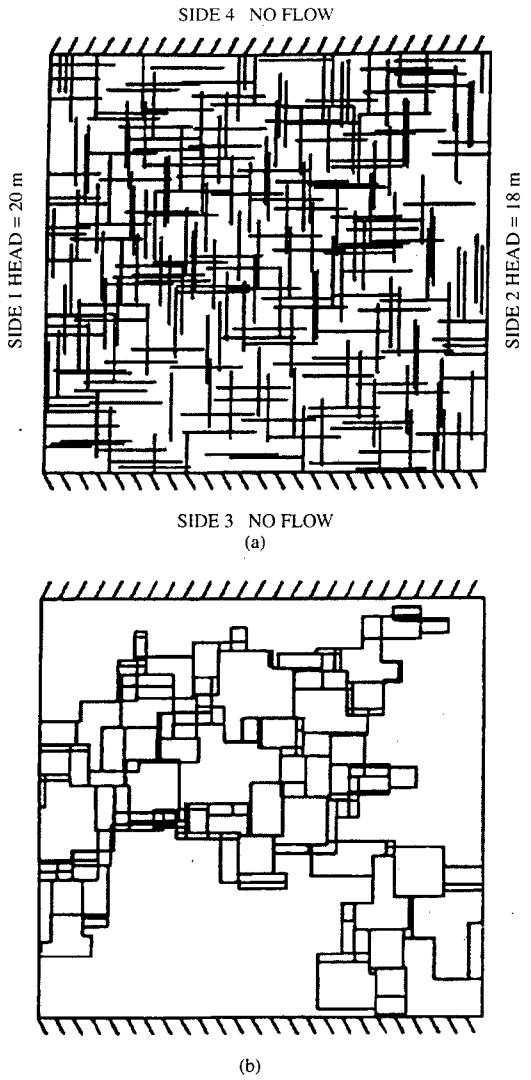


Fig. 1. (a) Original of the conductible network, and (b) Backbone of the fracture network in (a).

IV. METHODOLOGY

A two-dimensional and irregular fracture network with imposed boundary condition was considered as shown in Fig. 1. Flow was assumed only to be through the fractures. This model is similar to the one adopted by Smith and Schwarts (1984). The studied domain was arbitrarily limited to a square of $L_s \times L_s$=50 m×50 m with two orthogonal sets of fractures, which were numerically generated. A fracture was regarded as a "gap" between two parallel planes. The fractures were of equal length in each realization except that the fractures were truncated in the boundary. The fracture aperture was considered to be constant. The fracture spacing was represented as the average interval between every two adjacent fracture centers projecting on the boundary in a realization. One set of fractures was projected on the horizontal boundary and another set was projected on the vertical boundary. The hydraulic boundaries were arbitrarily designated at a constant head of 20 m for side 1, and a constant head of 18 m for side 2. No-flow boundaries were designated for side 3 and side 4 as shown in Fig. 1(a). Fig. 1(b) depicts a sample realization of the conductible networks labeled with boundary conditions. The detailed processes of generating fractures can be obtained from our previous study (Lee et al., 1994).

In order to investigate transport properties against fracture geometric parameters, fracture networks generating with several fracture spacings and fracture lengths were simulated. Transport properties in thirty realizations of conductible networks with a fracture spacing and a fracture length were averaged to achieve a basic statistic requirement. Flow rates through the fracture network were determined from the volumetric balance equations written for each node, requiring that the algebraic sum of the fluxes at a node equal zero. The Hagen-Poiseuille law was invoked for calculating of the rate of flow through a fracture. If we assume the domain has a unit thickness e, the flow rate in each fracture is given by

$$Q=e\rho g b^3 \Delta h/(12\mu_d L_f) \tag{8}$$

where ρ is the density of considered liquid , g is the gravity acceleration, b is the fracture aperture, L_f is the fracture length, μ_d is the dynamic viscosity of the fluid, and Δh is the head difference between the inlet and outlet of the fracture. In this paper b is assigned a value of 0.1 mm with no variation. A volumetric balance equation, written at each node, leads to a set of linear algebraic equations. This system is solved under the imposed boundary conditions, thereby providing pressure at each node in the network. Then, introducing the fracture apertures, the flow rate and fluid velocity were calculated in each fracture.

Injection of particles was performed at one of the boundary inlets that possessed a relative maximum flow rate. Two reasons to make such a choice are: (1) the extremely uneven flow leading to a concentration of the flow to certain preferred flow paths (Nordqvist et al., 1992) so that the particles were assumed to travel in the preferential path, i.e. the path of maximum flow rate; (2) the path with the greatest velocity was considered, and thus largest maximum flow rate caused the greatest velocity when the aperture was constant.

Following this solution, a random walk process was analyzed, governed by the flux regime in the network. Monte Carlo simulations were performed to track particles through the system, which simulated the movement of tracers carried by a flowing phase. The random walk process was based on the direction of flow and discharges in the fractures. When a particle left a fracture and entered a node, the adjoining fractures were examined to recognize whether or not their flow direction was away from the node. The particle then moved into one of these fractures with the probability of entry into each fracture proportional to its flow rate. This probability-aimed procedure is called the Monte Carlo simulation. Repeating this procedure enables the motion of a large number of particles to be tracked throughout the network. Computer simulations of the random walk were performed with 1000 particles tracked through the fracture system in each case.

In order to investigate dispersion coefficients, we needed to ensure that no particle passed through the fracture boundary. The following three procedures should be performed. Step one, particles are released from upstream boundary for all realizations. Step two, the travel time of particles arriving at the downstream boundary are recorded. Step three, the shortest one among the recorded times is selected and defined as the maximum duration for the particle to travel in each realization. The purposes of these procedures ensure no particle passing through the downstream boundary, and thus the dispersion coefficient can be estimated.

For the convenience of comparing between the dispersion coefficient and various fracture geometric parameters, a distance index RI (unit: length) is defined as

$$RI = X_b \times (T/\overline{t_{50}}) \tag{9}$$

where X_b is the length of domain in the direction of macroscopic hydraulic gradient, T is the time for particles to travel, t_{50} is the 50% break-through time of particles and $\overline{t_{50}}$ is the average time of 30 realizations of t_{50}. RI can be regarded as the average displacement in the x-direction. The x-direction is the direction of the macroscopic hydraulic gradient.

A computer code was written in **FORTRAN** to complete our work. Since the calculations were many and time-consuming, the program was run on a large computer station (VAX9420).

IV. RESULTS AND DISCUSSION

To investigate the dispersion phenomenon in saturated fracture networks, three mean fracture spacings (S=0.2 m, 0.30 m, 0.35 m) are generally selected. A network with relatively long fractures, small spacing or both, is called a "dense fracture structure". On the contrary, a network with relatively short fractures, large spacing or both, is called a "sparse fracture structure".

Figure 2 indicates that the slope values of $\ln\langle r^2\rangle$ vs. $\ln\langle t\rangle$ with various spacings (S=0.3 m, 0.35 m, 0.45 m, 0.5 m and 0.6 m) have a range between 1.27 and 1.66. A V-type track can be observed in each fracture spacing. A minimum value close to 1.27 exists, which closely matches the theoretical value of 1.27 provided by Sahami and Imdakm (1988). Meanwhile, increasing the fracture length with each fracture spacing decreases the slope value, and then after

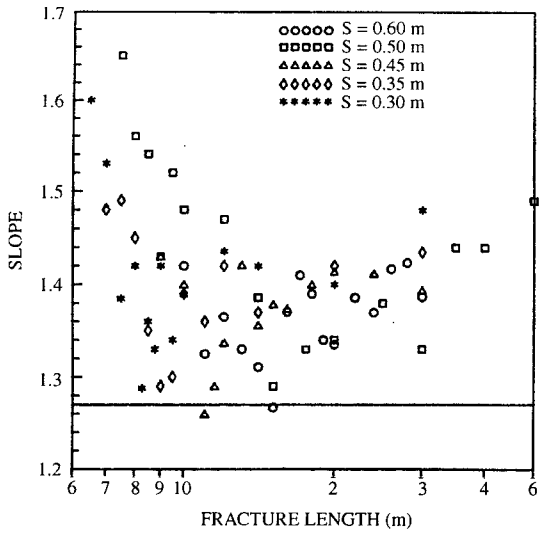


Fig. 2. Relationship between the slope values of $\ln\langle r^2\rangle$ vs. $\ln\langle t\rangle$ and mean fracture lengths for various mean fracture spacings S.

a minimum value of slope value, increasing the fracture length increases the slope value. This finding suggests that when the fracture length is extremely long (i.e. particles are transported only in a single fracture) or the fracture spacing is very small (i.e. particles are transported in very dense fracture network), the flow in the network becomes one-dimensional. Notably, one-dimensional transport has a slope value of 2 (Lin *et al.*, 1997).

Figure 3 represents the effect of fracture length on the anisotropic dispersion, illustrating the relation between D_{11} and RI. Figs. 3b, 3d and 3f show the relation between the standard deviation of D_{11} and RI. They indicate that D_{11} seems to linearly grow with the growth of RI as the standard deviation of D_{11} maintains an approximately constant value after particles migrate a specific distance. A fracture network with a longer fracture length has a larger D_{11} and a larger variation of D_{11}. A fracture network with a small fracture spacing seems to have a larger D_{11}, but this result is not clear. That is, a dense fracture structure could has a larger D_{11} as shown in Fig. 3.

Figures 4a-4f summarize the relation between D_{22} and RI in various assembles of fracture parameters. When RI is small, D_{22} has a trend to grow linearly with the growth of RI. However, D_{22} is likely to be constant when RI exceeds a certain length for the cases of the longer fracture lengths (e.g. Fig. 4c, L=14 m; Fig. 4e, L=10 m and L=14 m). As RI increases, the standard deviation of D_{22} decreases and then almost becomes a constant. This finding suggests that a dense fracture structure leads to a smaller value of D_{22}.

Figure 5 depicts the relation between the anisotropic ratio and RI with various fracture spacing. It indicates that the anisotropic ratio is small and nearly increased linearly with migration distance at early moving of the particles. In each case of Figs. 5a, 5b and 5c, the anisotropic ratio seems to remain constant or a small variation occurs after particles migrate a specified distance. This implies that there is an unique plume transport behavior after particles travel a long distance. By comparing Figs. 5a, 5b and 5c with the constant RI value, a dense fracture structure (i.e. small fracture spacing) leads to a larger D_{11}/D_{22}.

VI. CONCLUSION

This paper applies the discrete fracture approach to investigate the distance-dependent dispersion phenomenon in fractured networks. Under the imposed boundary conditions, the dispersion is a function of travel distance. The plume in a dense fracture structure is flatter since D_{11} is much larger than D_{22}. The anisotropic ratio of dispersion tensor is small and almost increases linearly with migration distance at early migration. The anisotropic ratio has a trend to remain constant or exhibit small variation after particles migrate a specified distance. In the dense fracture structure there is a larger D_{11}/D_{22}.

ACKNOWLEDGEMENT

The authors would like to thank the National Science Council (NSC) and the Radwaste Atomic Administration (RWA) for providing partial financial support of this study.

NOMENCLATURE

b	fracture aperture (L^{-1})
D_{ij}	dispersion coefficient tensor ($L^2\ T^{-1}$)
D_{11}	major principal dispersion coefficient ($L^2\ T^{-1}$)
D_{22}	minor principal dispersion coefficient ($L^2\ T^{-1}$)
e	domain thickness (L)
g	gravity acceleration ($L^2\ T^{-1}$)
L_s	length scale over which dispersion is studied (L)
L_f	fracture length (L)
L	mean length of fractures (L)
Δh	head difference across the fracture (L)
Q	fracture flow rate ($L^3\ T^{-1}$)
r	distance traveled by a particle (L)
RI	distance index (L)
S	mean fracture spacing (L)
T	maximum time for particles to travel (T)
t	time (T)
t_{50}	50% break-through time (T)
$\bar{t}_{50}$	average 50% break-through time (T)

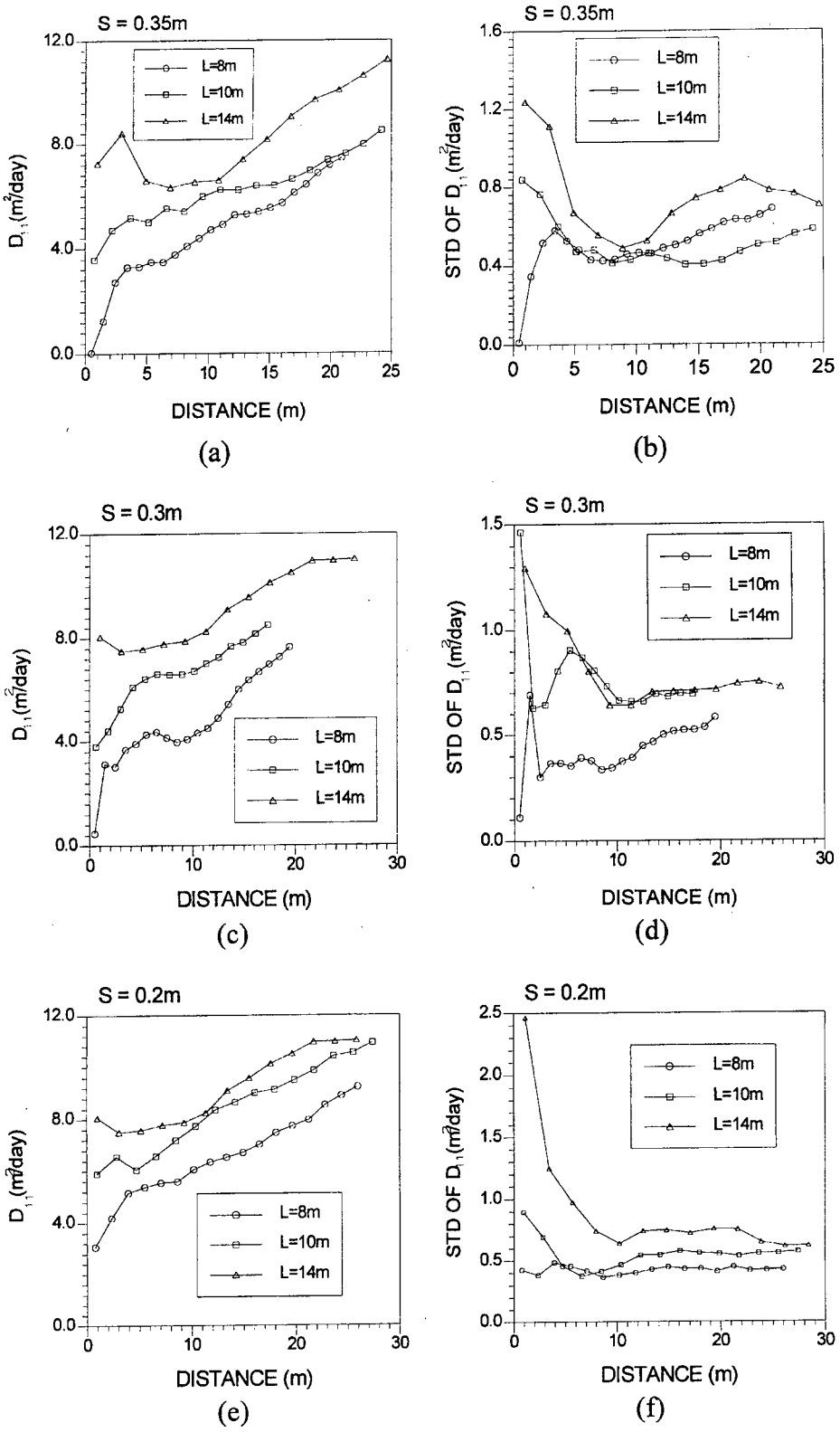


Fig. 3. D_{11} and its standard deviation as functions of distance index (RI), fracture spacing and fracture length.

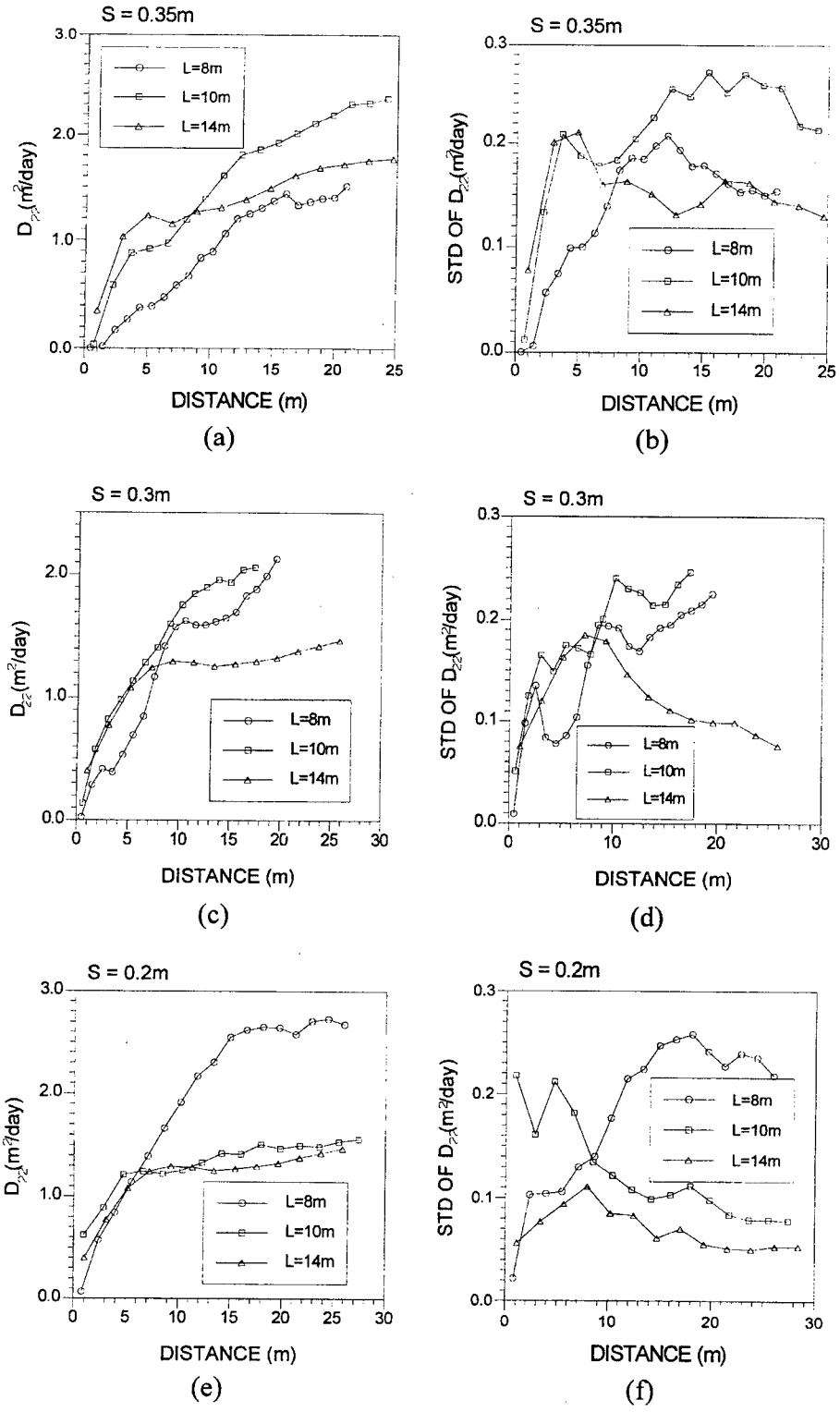


Fig. 4. D_{22} and its standard deviation as functions of distance index (RI) , fracture spacing and fracture length.

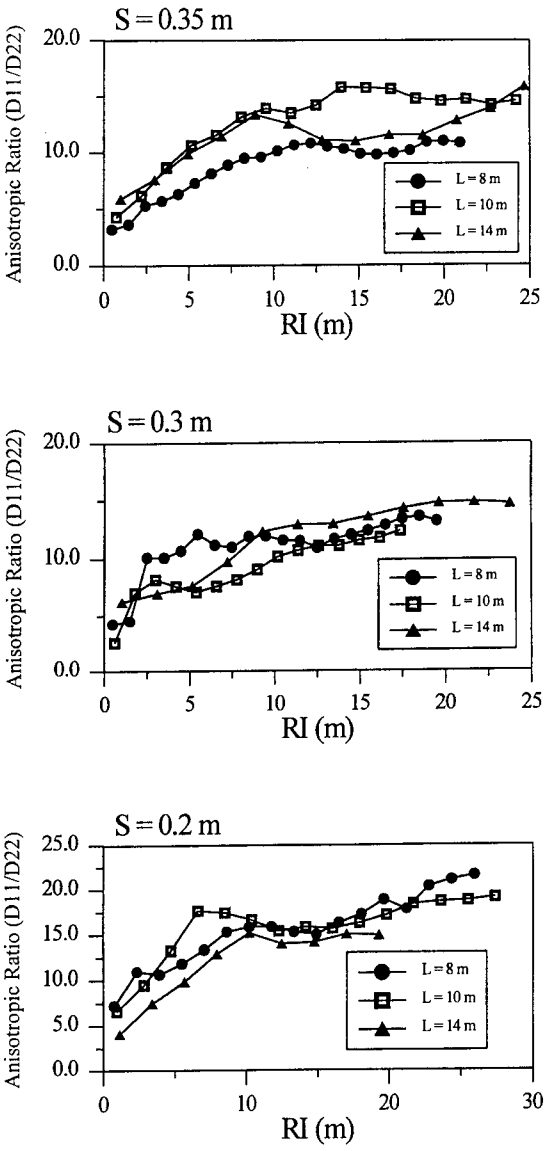


Fig. 5. Anisotropic ratio (D_{11}/D_{22}) as functions of distance index (RI) and fracture spacing.

μ_d dynamic fluid viscosity ($ML^{-1}T^{-1}$)
ρ density of fluid (ML^{-3})

< > average over a large population of particles
~ loose proportionality between left-hand side and right-hand side variables

REFERENCES

1. Berkowitz, B. and C. Braester, "Dispersion in Sub-representative Elementary Volume Fracture Network: Percolation Theory and Random Walk Approaches," *Water Water Resources Research*, Vol. 27, No. 12, pp. 3159-3164 (1991).
2. Lee, C.H., B.S. Lin and J.L. Yu, "Dispersion and Connectivity in Flow through Fractured Network," *Journal of the Chinese Institute of Engineers*, Vol. 17, No. 4, pp. 521-535 (1994).
3. Lin, B.S., C.H. Lee and H.H. Hwung, "Analysis of Mass Transport through Fracture Networks Using Percolation Theory Approach," *Journal of the Chinese Institute* of Environ. Engineering, Vol. 7, No. 1, pp. 1-11 (1997).
4. Nordqvist, A.W., Y.W. Tsang, C.F., Tsang , B. Dverstorp and J. Andersson, "A Variable Aperture Network Model for Flow and Transport in Fractured Rock," *Water Resources Research*, Vol. 28, No. 6, 1703-1713 (1992).
5. Sahimi, M., and A.O. Imdakm, "The Effect of Morphological Disorder on Hydrodynamic Dispersion in Flow through Porous Media," *J. Phys. A Math. Gen.*, Vol. 21, 3833-3870 (1988).
6. Schwartz, F.W. and L. Smith, "A Continuum Approach for Modeling Mass Transport in Fractured Media," *Water Resources Research*, Vol.24, No. 8, pp. 1360-1372 (1988).
7. Way, S.C. and C.R. McKee, : Restoration of In-Situ Coal Gasfication Sites from Naturally Occurring Flow and Dispersion, In-situ Consulting, Inc., Vol. 5, No. 2, pp. 77-101 (1981).

Discussions of this paper may appear in the discussion section of a future issue. All discussions should be submitted to the Editor-in-Chief.

Manuscript Received: July 28, 1997
Revision Received: Feb. 06, 1998
and Accepted: Feb. 19, 1998

破裂岩層中與距離相關之質點延散

林碧山　李振誥

國立成功大學資源工程

摘　要

本文主要以離散破裂面模式來探討質點於破裂岩石介質中與距離相關之延散現象。模擬結果顯示在遷移之早期，異向延散張量之比值隨遷移距離呈線性線增加而變大，但是在某距離之後有維持固定之傾向，同時，污染團呈橢圓狀擴散。

關鍵詞：透析理論、離散破裂面、延散。

Notes for Contributors

The Journal of the Chinese Institute of Engineers publishes original research papers on all phases of engineering and related fields in applied sciences. All papers submitted are referred by experts who will advise the Editor on the matter of acceptance on the basis of merit, originality and length of the paper. Only those papers that are highly recommended will be accepted, and this Journal reserves the right to accept the paper either as a full paper or as a short paper. A short paper is similar in form to a full paper, but, presents material in a brief nature or restricted scope. Four copies of manuscripts should be submitted to the Editor-in-Chief: Dr. Ching-Tien Liou, National Taiwan University of Science and Technology, No. 43, Sec. 4, Keelung Rd., Taipei, Taiwan, R.O.C. Manuscripts are reviewed with the understanding that the same work has not been and will not be published nor is presently submitted elsewhere. If the paper is accepted, the author will be responsible for proof-reading. There is a page charge of NT$500 per journal page. Twenty reprints of each paper are provided free of charge. For each extra reprint, the author will be charged NT$5 per page.

The Journal of the Chinese Institute of Engineers is an international journal. Foreign authors, when submitting a paper, can select an editor for reasons of geographic proximity (see the Board of Editors).

The author should indicate the submitted paper will be published as a full paper or as a short paper. Four copies of manuscripts are required for a full paper, and three for a short paper.

In addition to being concise and consistent in style, spelling and the use of abbreviations, the paper should conform to the following instructions.

1. Language: Papers should be written in English. Each paper should have abstracts (not exceeding 150 words) in both Chinese and English. For the author unfamiliar with Chinese, he may only submit the English abstract which will be translated into Chinese by the editor of this Journal.

2. Units: The SI units should be followed for all dimensional quantities.

3. Typescript: Manuscripts should be typed, double-spaced, on standard white paper measuring 21 cm 30 cm (A4 for mat).

4. Title & Author: The title of the paper should be concise, informafive and in capital letters. The author's name and affiliation should appear below the title.

5. Keywords: Several keywords (not more than 4 words) for the title of the paper should be given below the author's name and affiliation.

6. Headings: Headings for sections should be centered on the page with numbering. Headings for subsections should start from the left-hand margin.

7. Equations and Mathematical Formulas: All equations and mathematical formulas should be type-written or written clearly in ink. Equations should be numbered serially on the right-hand side by Arabic numerals in parentheses. Leave as much space as possible above and below all of the mathematical expressions.

8. Length: The text of a full paper normally should not exceed 8 printed pages in the Journal or equivalent. A short paper normally should not exceed 4 pages or equivalent.

9. Figures: Once the paper is accepted, the author should promptly supply original copies of all of the illustrations, line drawings drafted in ink on white or drawing paper (or "glossies"), and photographs on glossy papers. All illustrations, photographs, tables, etc. should be numbered, titled and have descriptive captions. The figure number and author's name should be clearly indicated on the reverse side of each illustration. The lettering, typed or in engineering type, should be large enough so that when reduced, it will still be legible.

10. Nomenclature: All symbols and units should be listed in English and in alphabetical order with indication of their meaning and dimensions.

11. References: References should be listed and numbered in alphabetical order according to author, Patentee, or editor. Complete information with the title of the paper, Patent, report, etc. should be given. Examples:

1. Chu, S. and C.S. Wang, "TITLE," CSITR-668-72, Chung Shan Institute of Science and Technology, Lungtan, Taiwan (1977).
2. Etkin, B., Dynamics of Atmospheric Flight, John Wiley and Sons, New York, N.Y., pp. 166-188 (1970).
3. Hsiao, C.H., "TITLE," Ph.D. Thesis, Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan (1974).
4. Morris, J.G. and K.K. Howard, "Thermomechanical Treatments of Alloys," *Journal of Applied Physics*, Vol. 42, No. 1, pp. 32-325 (1971).
5. Pfaltz, J.L. and A. Rosenfeld, "TITLE," *Proc. of First Intemational Joint Conference on Artificial Intelligence*, Washington, D.C. (1969).

作者注意事項（84年4月修訂）

本學刊出版工程及相關應用科學範圍之原創性論文。所收稿件將送所屬學門之專家審核，按文章之水準及長度向總編輯作適當之推薦。本學刊得以論文或短篇論文形式刊出，短篇論文之形式與論文相似，但其內容較為簡短，稿件及通訊請寄學刊總編輯：台北市基隆路四段43號國立台灣科技大學劉清田校長。凡投寄稿件必須不在其他期刊考慮出版者。稿件如經接受，校對工作將由作者負責。作者需付本學刊論文刊登費每頁新台幣伍佰元。本學刊贈送作者論文抽印本二十份；若作者欲索取額外的份數，則每份每頁酌收印刷工本費新台幣伍元。所付費用均將開具正式收據。

本學刊係國際化刊物，海外作者如欲投稿可選擇就近編輯學者擲寄文稿（請參考編輯委員名單）。

作者投稿時請註明以「論文」或「短文」形式刊出，以「論文」形式投稿者，請備妥稿件四份，「短文」者三份。

稿件宜簡潔，且合乎下列格式：

1. **文字**　稿件應以英文撰寫，並附不超過150字之中文及英文摘要。不諳中文之作者其文稿之摘要將由本刊代為中譯。
2. **單位**　所有含因次之量須採用SI單位。
3. **打字**　稿件必須用白紙打字，列與列間需有一字之間隔，用紙大小以21公分×30公分（A4規格）為準。
4. **題目、作者**　論文題目宜簡明，英文題目應以大寫字體打印。作者姓名及服務機關並列於論文題目之下方。
5. **關鍵詞**　在題目中須選出二至四個關鍵詞，並置於作者姓名及服務機關之下方。
6. **章節及小節標題**　論文之章節標題須列於稿紙之中央對稱位置，且加編號。小節標題必須從文稿之左緣開始。
7. **數學式**　所有公式及方程式均須打字或以黑墨書寫清楚，其後標明式號於圓弧括內。為清晰起見，每一式之上下須多空一列。
8. **長度**　論文之長度以不超過學刊8頁或其相當之長度為準；短篇論文以不超過學刊4頁或其相當之長度為準。
9. **插圖**　稿件經通知採用後，如有插圖、照片，作者應迅速提供白紙或描圖紙上墨繪製之圖表，黑白光面照片等原件。所有圖表、照片必須附有編號及標題或簡短說明，用鉛筆註明作者姓名；字體一律以打字或工程字體為準，且須夠大，其大小之決定原則為原圖縮小成橫寬8公分時圖中字體符號高2毫米。
10. **符號說明**　論文後須有符號說明及其單位，並按英文字母先後次序排列。
11. **參考文獻**　所有參考文獻須按作者姓氏之英文字母先後次序排列且加編號。論文中引用到參考文獻之寫法請見前頁。